

Best Available Copy

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

#2 P/1139-108  
J1036 U.S. PTO  
10/010330  
11/08/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年11月 9日

出 願 番 号

Application Number:

特願2000-341680

出 願 人

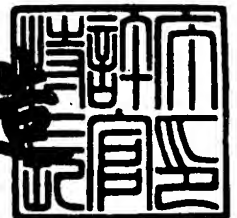
Applicant(s):

日本電気株式会社

2001年 8月31日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



CERTIFIED COPY OF  
PRIORITY DOCUMENT

出証番号 出証特2001-3077796

【書類名】 特許願

【整理番号】 33509803

【提出日】 平成12年11月 9日

【あて先】 特許庁長官 殿

【国際特許分類】 H04L 13/00

【発明の名称】 データ通信システム、その通信方法及びその通信プログラムを記録した記録媒体

【請求項の数】 11

【発明者】

    【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

    【氏名】 竹田 憲司

【特許出願人】

    【識別番号】 000004237

    【氏名又は名称】 日本電気株式会社

【代理人】

    【識別番号】 100108578

    【弁理士】

    【氏名又は名称】 高橋 詔男

【代理人】

    【識別番号】 100064908

    【弁理士】

    【氏名又は名称】 志賀 正武

【選任した代理人】

    【識別番号】 100101465

    【弁理士】

    【氏名又は名称】 青山 正和

【選任した代理人】

    【識別番号】 100108453

    【弁理士】

【氏名又は名称】 村山 靖彦

【手数料の表示】

【予納台帳番号】 008707

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9709418

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ通信システム、その通信方法及びその通信プログラムを記録した記録媒体

【特許請求の範囲】

【請求項 1】 データ送信装置とデータ受信装置がネットワークで接続され、前記データ送信装置は前記データ受信装置へデータパケットを送信し、前記データ受信装置は前記データ送信装置に受信したデータパケットの送達確認を通知する ACK パケットを送信し、前記データ送信装置は前記 ACK パケットの内容から送信したデータパケットの損失を検出し、該損失を検出したデータパケットを再送することによりデータパケットのエラー制御を行なうデータ通信システムにおいて、

前記データ送信装置は、

送信するデータパケットのラウンドトリップ数を計数するカウンタと、

送信した各データパケットに対して、該データパケットと該データパケット送信時の前記カウンタの値の対応を記憶し、前記記憶したカウンタの値がその時点での前記カウンタの値より 2 つ以上小さい場合に、該記憶したカウンタ値に対応するデータパケットが損失したとみなし、該データパケットを再送する手段と

を備えることを特徴とするデータ通信システム。

【請求項 2】 前記データ送信装置において、送信するデータパケットにその時点での前記カウンタ値をラウンドトリップ通知情報として含ませる手段と、

前記データ受信装置において、受信したデータパケットに対する ACK パケットに該受信したデータパケットがラウンドトリップ通知情報として含む前記カウンタ値をラウンドトリップ応答情報として含ませる手段と、

前記データ送信装置において、受信した ACK パケットにラウンドトリップ応答情報として含まれている値が前記カウンタの値と等しい場合に、前記カウンタの値を 1 つ増加させる手段と

をさらに備えることを特徴とする請求項 1 記載のデータ通信システム。

【請求項 3】 データ送信装置とデータ受信装置がネットワークで接続され、前記データ送信装置は前記データ受信装置へデータパケットを送信し、前記デ

ータ受信装置は前記データ送信装置に受信したデータパケットの送達確認を通知するACKパケットを送信し、前記データ送信装置は前記ACKパケットの内容から送信したデータパケットの損失を検出し、該損失を検出したデータパケットを再送することによりデータパケットのエラー制御を行なうデータ通信システムにおいて、

前記データ送信装置は、

送信するデータパケットのラウンドトリップ数を計数するカウンタと所定のデータを記憶する第1及び第2のテーブルと、

送信した各データパケットに対して、該データパケットと該データパケット送信時の時刻の対応を前記第1のテーブルに記憶する手段と、

前記カウンタの取る各カウンタ値に対して、カウンタ値と前記カウンタがそのカウンタ値を取った後初めてデータパケットを送信するときの送信時の時刻との対応を前記第2のテーブルに記憶する手段と、

前記第1のテーブルに記憶した時刻の値が、前記第2のテーブルに記憶したその時点での前記カウンタの値より2つ小さいカウンタ値に対応する時刻より早い場合に、該第1のテーブルに記憶した時刻に対応するデータパケットが損失したとみなし、該データパケットを再送する手段と

を備えることを特徴とするデータ通信システム。

【請求項4】 前記データ送信装置において、送信するデータパケットにその時点での時刻を時刻通知情報として含ませる手段と、

前記データ受信装置において、受信したデータパケットに対するACKパケットに該受信したデータパケットが時刻通知情報として含む前記時刻を時刻応答情報として含ませせる手段と、

前記データ送信装置において、受信したACKパケットに時刻応答情報として含まれている時刻が、前記第2のテーブルに記憶したその時点での前記カウンタの値と対応させて記憶させた時刻と等しいかまたは大きい場合に、前記カウンタの値を1つ増加させる手段と

をさらに備えることを特徴とする請求項3記載のデータ通信システム。

【請求項5】 データ送信装置とデータ受信装置がネットワークで接続され

、前記データ送信装置は前記データ受信装置へデータパケットを送信し、前記データ受信装置は前記データ送信装置に受信したデータパケットの送達確認を通知するACKパケットを送信し、前記データ送信装置は前記ACKパケットの内容から送信したデータパケットの損失を検出し、該損失を検出したデータパケットを再送することによりデータパケットのエラー制御を行ない、更には、データパケットの損失を検出後、該データパケットの送達がデータ送信装置によって確認されるまでのエラー制御中に送信ウィンドウにより送信フロー制御を行なうデータ通信システムにおいて、

前記データ送信装置は、エラー制御中に前記データ受信装置から受信したACKパケットで新規に送達確認がなされたデータの合計サイズ分の閉じられたウィンドウを前記送信ウィンドウから解放し、その分を送信可能とする

ことを特徴とするデータ通信システム。

【請求項6】 データ送信装置とデータ受信装置がネットワークで接続され、前記データ送信装置は前記データ受信装置へデータパケットを送信し、前記データ受信装置は前記データ送信装置に受信したデータパケットの送達確認を通知するACKパケットを送信し、前記データ送信装置は前記ACKパケットの内容から送信したデータパケットの損失を検出し、該損失を検出したデータパケットを再送することによりデータパケットのエラー制御を行なうデータ通信方法において、

前記データ送信装置は、送信するデータパケットのラウンドトリップ数を計数するカウンタを保持し、

送信した各データパケットに対して、該データパケットと該データパケット送信時の前記カウンタの値の対応を記憶し、前記記憶したカウンタの値がその時点での前記カウンタの値より2つ以上小さい場合に、該記憶したカウンタ値に対応するデータパケットが損失したとみなし、該データパケットを再送する

ことを特徴とするデータ通信方法。

【請求項7】 前記データ送信装置は、送信するデータパケットにその時点での前記カウンタ値をラウンドトリップ通知情報として含ませ、

前記データ受信装置は、受信したデータパケットに対するACKパケットに該受信したデータパケットがラウンドトリップ通知情報として含む前記カウンタ値を

ラウンドトリップ応答情報として含ませ、

前記データ送信装置は、受信したACKパケットにラウンドトリップ応答情報として含まれている値が前記カウンタの値と等しい場合に、前記カウンタの値を1つ増加させる

ことを特徴とする請求項6記載のデータ通信方法。

【請求項8】 データ送信装置とデータ受信装置がネットワークで接続され、前記データ送信装置は前記データ受信装置へデータパケットを送信し、前記データ受信装置は前記データ送信装置に受信したデータパケットの送達確認を通知するACKパケットを送信し、前記データ送信装置は前記ACKパケットの内容から送信したデータパケットの損失を検出し、該損失を検出したデータパケットを再送することによりデータパケットのエラー制御を行なうデータ通信方法において、前記データ送信装置は、

送信するデータパケットのラウンドトリップ数を計数するカウンタと所定のデータを記憶する第1及び第2のテーブルを保持し、

送信した各データパケットに対して、該データパケットと該データパケット送信時の時刻の対応を前記第1のテーブルに記憶し、

前記カウンタの取る各カウンタ値に対して、カウンタ値と前記カウンタがそのカウンタ値を取った後初めてデータパケットを送信するときの送信時の時刻との対応を前記第2のテーブルに記憶し、

前記第1のテーブルに記憶した時刻の値が、前記第2のテーブルに記憶したその時点での前記カウンタの値より2つ小さいカウンタ値に対応する時刻より早い場合に、該第1のテーブルに記憶した時刻に対応するデータパケットが損失したとみなし、該データパケットを再送する

ことを特徴とするデータ通信方法。

【請求項9】 前記データ送信装置は、送信するデータパケットにその時点での時刻を時刻通知情報として含ませ、

前記データ受信装置は、受信したデータパケットに対するACKパケットに該受信したデータパケットが時刻通知情報として含む前記時刻を時刻応答情報として含ませ、

前記データ送信装置は、受信したACKパケットに時刻応答情報として含まれている時刻が、前記第2のテーブルに記憶したその時点での前記カウンタの値と対応させて記憶させた時刻と等しいかまたは大きい場合に、前記カウンタの値を1つ増加させる

ことを特徴とする請求項8記載のデータ通信方法。

【請求項10】 データ送信装置とデータ受信装置がネットワークで接続され、前記データ送信装置は前記データ受信装置へデータパケットを送信し、前記データ受信装置は前記データ送信装置に受信したデータパケットの送達確認を通知するACKパケットを送信し、前記データ送信装置は前記ACKパケットの内容から送信したデータパケットの損失を検出し、該損失を検出したデータパケットを再送することによりデータパケットのエラー制御を行ない、更には、データパケットの損失を検出後、該データパケットの送達がデータ送信装置によって確認されるまでのエラー制御中に送信ウィンドウにより送信フロー制御を行なうデータ通信方法において、

前記データ送信装置は、エラー制御中に前記データ受信装置から受信したACKパケットで新規に送達確認がなされたデータの合計サイズ分の閉じられたウィンドウを前記送信ウィンドウから解放し、その分を送信可能とする

ことを特徴とするデータ通信方法。

【請求項11】 請求項6～10のいずれか1項記載のデータ通信方法をコンピュータを用いて実行する際に用いられるプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データ通信装置、データ通信システム、その通信方法及びその通信プログラムを記録した記録媒体に関し、特に、OSI (Open System Interconnection) 参照モデル第4層プロトコルやインターネット・トランスポート層プロトコルにおいて再送によるパケット転送誤り制御を行なう際に用いて好適なデータ通信装置、データ通信システム、その通信方法及びその通信プログラムを記録した

記録媒体に関する。

【 0 0 0 2 】

【従来の技術】

現在、データ通信インフラとして最も普及しているインターネットにおいて、米国Internet Engineering Task Forceの“Transmission Control Protocol”、Request For Comments（以降RFCと記述する）793（引用文献A）と、同、“Requirements for Internet Hosts — Communication Layers”、RFC1122で規定される通信プロトコルTCP（Transmission Control Protocol）は信頼性を保証したデータ配送サービスを提供するトランスポート層プロトコルとして広く使用されている。

【 0 0 0 3 】

TCPでは信頼性を保証するために、データ受信側TCP処理部（受信側TCPと呼ぶ）によるデータ送信側TCP処理部（送信側TCPと呼ぶ）への累積確認メッセージ（ACK（Acknowledge；肯定応答）メッセージ）通知による累積確認通知機構と、送信側TCPによって実行される、通知されたACKメッセージに基づく送達確認と、更に送信側TCPによる再送タイマによるデータセグメントの損失検出・再送信を行なうエラー制御機構を備えている。送信側TCPは受信側TCPから通知されるACKメッセージにより、データの送信開始以来、抜けなく受信側TCPに到着しているデータの最後のデータバイトの番号を知ることができる。これにより、既に送信したデータセグメントの受信側TCPへの送達を確認することができる。また、送信側TCPは、データセグメント送信時に再送タイマを起動し、再送タイマがタイムアウトするまでにデータセグメントの送達確認が取れない場合、そのデータセグメントは損失したとみなしデータセグメントを再送する。ただし、TCPでは、データセグメントの損失の主な原因はネットワーク内の輻輳の発生と仮定している。そのため、損失したセグメントの再送およびその後のデータセグメント送信の際にはデータセグメント損失の原因となったネットワーク内の輻輳を緩和すべくデータセグメント送信速度の抑制、すなわち、輻輳制御を行なう。RFC793とRFC1122で規定されるTCPにはSlow Start（スロー・スタート）とCongestion Avoidance（輻輳回避）と呼ばれる二つの輻輳制御の機構が定義されている。

## 【 0 0 0 4 】

Slow Startは再送タイマがタイムアウトしたときに適用される制御で、送信側TCPが輻輳ウィンドウ (Congestion Window: cwnd) と呼ばれる送信ウィンドウのサイズを最大セグメントサイズ (Maximum Segment Size: MSS) 一つ分のサイズに設定して、それまで累積確認されているデータの次のデータから送信 (再送) を開始する。そして、以降送信したデータセグメントの送達確認が取れるたびにcwndを1MSS分のサイズだけ増やす制御である。Congestion AvoidanceはSlow Startにおけるcwndのサイズが再送タイマのタイムアウト時点でのcwndサイズの半分を超えた時点から適用される制御で、cwndのサイズ分のデータセグメントの送達確認が取れるたびにcwndのサイズを1MSS分増やす制御である。送信側TCPが受信側からの送達確認の取れていないデータを送る際の上限値はcwndの値と受信側TCPからTCPヘッダ中のWindow Size (ウィンドウサイズ) フィールドで通知される値の小さい方で規定されるため、受信側TCPから通知されるWindow Sizeフィールドの値が十分大きい場合はcwndの値が実質の送信ウィンドウサイズとなり、Slow StartとCongestion Avoidanceアルゴリズムによる制御がTCPのスループット性能に対して支配的になる。そのため、このようなWindow Sizeフィールドの値が十分大きい場合にデータセグメントが損失したときのTCPの動作は、送信側TCPは再送タイマのタイムアウトによりその損失を検出し、cwndをデータセグメント一つ分のサイズに縮小して損失が検出されたデータセグメントを再送し、その時点でのトラヒック投入速度を一時的に大きく落とし、以降、送信側TCPはcwndのサイズが、再送タイマがタイムアウトした時点でのcwndサイズの半分になるまでSlow Startアルゴリズムによりcwndのサイズを指数関数的に急速に増加させてトラヒック投入速度を急速に増加させるが、その後はCongestion Avoidanceアルゴリズムによりスループットは比較的緩やかにしか回復しない。このようにデータセグメント損失時にスループットの抑制速度より回復速度を遅くするのは、本制御によりネットワークが輻輳状態と非輻輳状態を繰り返すような不安定な状態となることを回避するためである。

## 【 0 0 0 5 】

Slow StartおよびCongestion Avoidanceは、データセグメントの損失の検出に

時間がかかることと、cwndサイズの回復に時間がかかることから、TCPのスループット性能向上のために、データセグメント損失後のエラー制御の効率化、すなわち、送信側TCPによるデータセグメント損失の検出時間の短縮やより効率的な再送制御方法等の開発がなされてきた。

#### 【 0 0 0 6 】

TCP Renoと呼ばれるTCPの一実装においては、“TCP Congestion Control”、RFC2581（引用文献B）で示されるFast Retransmit & Fast Recovery（ファストリトランスミット・ファストリカバリ：以降、FR&FRと呼ぶ）と呼ばれる技術が導入された。Fast Retransmit（以降、FRetと呼ぶ）では、損失したデータセグメントの後続のデータセグメント受信によって受信側TCPから送信される“内容が同じであるACKセグメント”（Duplicate ACK：重複ACKと呼ぶ）を送信側TCPが規定数受信することにより、送信側TCPが当該重複ACKが送達確認を示すデータの次のデータセグメントが損失したことを検出し、即座に損失したデータセグメントを再送する。ここで、送信側TCPが重複ACK受信時に即座に再送せずに規定数の重複ACKを受信するまで待つのは、送信したデータセグメントの受信側TCPでの到着順序が入れ変わった場合に受信側TCPが送信する重複ACKで誤って不要なデータセグメントの再送を行なう確率を低めるためである。また、Fast Recovery（以降、FRecと呼ぶ）では、送信側TCPにおける重複ACK受信は送信したデータセグメントが受信側TCPに受信されたことを示すことから、送信側TCPは、重複ACK受信による再送後、重複ACKを受信するとデータセグメント1つ分送信ウィンドウを解放し、それにより新たにデータセグメントを送信する余地ができれば後続のデータセグメントを送信する。これを再送したデータセグメントの送達確認が取れるまでのエラー制御処理中に適用する。但し、このとき送信側TCPはネットワークの輻輳緩和のために実質的な送信ウィンドウが再送時のcwndの値の半分に保たれるように制御する。また、再送したデータセグメントの送達確認が取れたらFRecのエラー制御を終了し、cwndの値が前記再送時のcwndの値の半分からCongestion Avoidanceを開始する。

#### 【 0 0 0 7 】

上記FR&FRは送信側TCPが連続的にデータセグメントを送信している際に一つ

の送信ウィンドウサイズ内で送信したデータセグメントの内の一つのデータセグメントが損失した場合に有効である。この場合、上記の再送タイマとSlow Start、Congestion Avoidanceによる再送制御および輻輳制御に比べ、データセグメントの損失を早期に検出し、その損失したデータセグメントを早期に再送し、かつ、cwndサイズを比較的大きな値に保ったまま再送制御を終えて通常の通信状態に復帰するためTCPのスループット性能が改善される。

## 【 0 0 0 8 】

しかし、上述のFast Retransmit & Fast Recoveryでは、1つの送信ウィンドウ内で送信されたデータセグメントが複数個損失した場合、各損失したデータセグメントによるFast Retransmitの起動が複数回発生する場合もあり、この場合はcwndの値がそれぞれ元の値の半分に縮小され、cwndの値が小さくなってしまう。また、各データセグメントの損失において、その後、送信側TCPは規定数の重複ACKを受信できない場合には再送タイマのタイムアウトによる損失の検出が必要となり、それに続くSlow StartとCongestion AvoidanceによりTCPのスループット性能がそれほど改善されない場合がある。この問題に対し、“The NewReno Modification to TCP ‘s Fast Recovery Algorithm”、RFC2582（引用文献C）で示されるFast Recoveryの改良と、“TCP Selective Acknowledgment Options”、RFC2018（引用文献D）で示されるTCP SACK（Selective Acknowledgments）オプションの適用の二つの解決策が示されている。

## 【 0 0 0 9 】

引用文献Cで示されるFast Recoveryの改良は、それまでのFast Retransmitでは送信側TCPがエラー制御処理中、すなわち、Fast Recovery中にそれまでとは内容の異なる重複ACKを規定数受信した場合には再度Fast Retransmit & Fast Recoveryを実行していたのに対して、本方式では同様な状況において送信側TCPはFast Retransmitは実行せず、それまでの重複ACKと内容の異なる（それまでの重複ACKで通知されていた送達確認データセグメントが更新された）ACK（partial ACKと呼ぶ）の受信に対して更新された送達確認データセグメントの次のデータセグメントが損失したことを検出し、検出した損失セグメントの再送を行なう。

## 【 0 0 1 0 】

引用文献Dで示されるTCP SACKオプションは、受信側TCPが受信したデータセグメントの累積確認に加え、累積確認したデータセグメント以降の受信データセグメントを通知する。送信側TCPはこの情報を利用することにより、早期のデータセグメント損失の検出や無駄なデータセグメント再送を回避し、スループット特性向上を図ることができる。

#### 【 0 0 1 1 】

引用文献Cで示されるFast Recoveryの改良の動作ロジックは、TCP NewRenoと呼ばれるTCPの一実装形態の中に実装されている。また、TCP SACKオプションは、引用文献Dにオプションフィールドの構成等は記述されているが、SACKオプションを受信した場合の送信側TCPの動作は実装依存となっており様々な実装が存在している。Fall, K. and Floyd, S. "Simulation-based Comparison of Tahoe, Reno and SACK TCP", 米国Association for Computing Machinery (ACM) Computer Communication Review V. 26 N. 3 pp. 5-21、1996年7月(引用文献E)に幾つかの実装方式が記載されている。

#### 【 0 0 1 2 】

ここでは、図13から図18を参照して、前記引用文献Eに記述されているSACKオプションを用いたTCPの技術を説明する。

#### 【 0 0 1 3 】

図13に示されるように、既存の技術におけるシステムは送信装置1、受信装置2、ネットワーク3から構成され、送信装置1と受信装置2はネットワーク3を介して接続される。送信装置1は、アプリケーションサービス実行手段11とデータセグメント送信処理手段12、ACK受信処理手段13、パケット送信手段14、パケット受信手段15から構成される。受信装置2は、アプリケーションサービス実行手段21とACK送信処理手段22、データセグメント受信処理手段23、パケット送信手段24、パケット受信手段25から構成される。

#### 【 0 0 1 4 】

アプリケーションサービス実行手段11とアプリケーションサービス実行手段21は、対になって利用者にサービスを提供する手段であり、基本的にアプリケーションサービス実行手段11からアプリケーション実行手段21にデータを転送するこ

とによりサービスを実現する。データの流れる方向によりそれぞれ役割が変わるが、Web（ウェブ）ブラウザとWebサーバの対や、FTP（File Transfer Protocol）クライアントとFTPサーバの対などがこれに相当する。パケット送信手段14とパケット送信手段24は、それぞれ、上位から渡されたTCPセグメントをネットワーク3を通じて相手側に送信するものであり、IP（Internet Protocol）プロトコルスタックとMAC（Media Access Control）ドライバ、NIC（Network Interface Card）の集合体に相当する。パケット受信手段15とパケット受信手段25は、それぞれ、ネットワーク3を通じて相手側から送られてきたパケットを受信するものであり、同様に、IPプロトコルスタックとMACドライバ、NICの集合体に相当する。

#### 【 0 0 1 5 】

ネットワーク3はEthernet（イーサネット）などのデータリンクとルータで構成されたIPネットワークである。データセグメント送信処理手段12とACK受信処理手段13、ACK送信手段22、データセグメント受信処理手段23はアプリケーションサービス実行手段11から渡されるデータをアプリケーション実行手段21に誤りなく渡すためのプロトコル処理を行なうものであり、データセグメント送信処理手段12とACK受信処理手段13はTCPのデータセグメント送信処理部に相当し、ACK送信処理手段22とデータセグメント受信処理手段23はTCPのデータセグメント受信処理部に相当する。

#### 【 0 0 1 6 】

データセグメント送信処理手段12は、アプリケーションサービス実行手段11から渡されるデータを送信バッファに蓄積し、送信バッファに蓄積されているデータからTCPセグメントを生成してパケット送信手段14に渡すことを基本機能とし、さらに、輻輳制御とフロー制御の一端とSACKオプションを利用した場合のエラー回復制御の一端を備える。データセグメント送信処理手段12が備える輻輳制御とフロー制御の一端とは、輻輳制御とフロー制御におけるTCPセグメント送信の可否を判断することである。引用文献Aに示されているとおり、TCPでは、（1）累積確認されたデータの1バイト後続のデータのシーケンス番号を指すSND、UNA（以降snd\_unaと記述する）、（2）次に送信すべきデータの先頭のデータのシ

ーケンス番号を指すSND. NXT（以降snd\_nxtと記述する）、（3） 相手側からTCPヘッダのWindowフィールドで通知された相手側の受信ウィンドウサイズを指すSND. WND（以降snd\_wndと記述する）、また、引用文献Aには記載されていないが、輻輳制御のための送信ウィンドウである輻輳ウィンドウ（以降snd\_cwndと記述する）の各情報を管理している。データセグメント送信処理手段12は、送達確認が取れていない送信済みデータの合計サイズが相手側受信ウィンドウsnd\_wndと輻輳ウィンドウsnd\_cwndのいずれよりも小さい場合にのみ後続のデータからデータセグメントを生成し、送信する。

#### 【0017】

データセグメント送信処理手段12が備えるSACKオプションを利用したエラー回復制御の一端とは、エラー回復処理中のデータセグメント送信の可否の判断と再送データセグメントの選定、エラー回復処理中のネットワークに滞留している転送中のデータの推定値（以降、pipeと記述する）の値の管理である。引用文献Eによると、データセグメント送信処理手段12は、エラー回復処理中には滞留データ推定値pipeが輻輳ウィンドウsnd\_cwndを下回るときのみその差分を最大サイズとしてデータを送信する。データを送信する際は、累積確認とSACKオプションで送達確認が通知されていないデータの内、再送していないデータをシーケンス番号が小さい順に選択して再送する。再送するデータが存在しない場合には未送信の後続のデータを送信する。いずれにおいても、データセグメント送信後は送信したデータセグメントのサイズを滞留データ推定値pipeに加算する。

#### 【0018】

また、引用文献Eでは、送信側TCPは受信側TCPからSACKオプションで通知されたデータセグメント送達情報をスコアボードと呼ばれるリストで管理している。スコアボードの一実装例としては図14で示されるものがある。スコアボードは送達確認が取れたデータの塊（データブロック）の先頭データのシーケンス番号と末尾データのシーケンス番号を格納（それぞれ図14中、“start”と“end”）したリスト要素を、各自が格納しているデータブロックのシーケンス番号の順番に並べたリスト構造になっている。各リスト要素では更に一つ前のリスト要素の表すデータブロックから自リスト要素の表すデータブロックまでの内、既に再送

されたデータの最後のデータのシーケンス番号を格納している（図14中、“retran”）。これにより次に再送すべきデータがどれなのかがわかるようになっている。

## 【 0 0 1 9 】

データセグメント受信処理手段23は、パケット受信手段25から渡されるデータセグメント送信処理手段12から送信されたデータセグメントを受信し、受信したデータを順序正しく受信バッファに格納する。また、受信したデータセグメントの情報をACK送信処理手段22に通知する。

## 【 0 0 2 0 】

ACK送信処理手段22はデータセグメント受信処理手段23から通知されるデータセグメントの情報からACKメッセージを作成し、パケット送信手段24に渡す。ACK送信処理手段22は、これまで受信したデータセグメントが含むデータに抜けがあり、幾つかのデータブロックが存在する場合、引用文献Dに記述されている方法でSACKオプションを用いてデータブロックを通知する。

## 【 0 0 2 1 】

ACK受信処理手段13はパケット受信手段15から渡されるACK送信手段22が送信したACKメッセージを処理する手段で、更に、SACKオプション処理手段131とACK処理手段132から構成される。SACKオプション処理手段131は、パケット受信手段15から渡されたACKメッセージにSACKオプションが付いている場合、その内容をスコアボードに反映させる。ACK処理手段132は、エラー制御処理中でない場合、すなわち、Slow StartやCongestion Avoidance中は重複ACKを監視し、重複ACKが規定回数（以降、規定回数をrxmtthreshとも記述する）連続で受信された場合は、前記の通り、FRetの処理を行なう。その際には、エラー制御処理中の場合の処理のため初期化処理として、その時点で送信済みであるデータの最後のデータのセグメント番号（以降、snd\_maxと記述する）を記憶し（以降、記憶したセグメント番号をp\_max\_seqで参照する）、推定した滞留データサイズをpipeに設定し、その時点での輻輳ウィンドウの半分の値を記憶し（以降、記憶した値をssthreshで参照する）、輻輳ウィンドウsnd\_cwndを半分にする。

## 【 0 0 2 2 】

ACK処理手段132は、エラー制御処理中の場合、(1) 重複ACKに対してはpipeの値を1MSSサイズ分小さくし、(2) 累積確認が更新された場合には、累積確認されたデータを送信バッファから削除し、その内容をスコアボードに反映し(累積確認されたデータの情報を格納するリスト要素をスコアボードから削除)、snd\_unaをACKメッセージの通知する累積確認の値(以降、ti\_ackと記述する)に更新する。その後、上記、(1)、(2) いずれの場合もデータセグメント送信処理手段11に送信要求を行なう。FRet起動時に送信していたデータセグメントが全て送達確認されたら、すなわち、ti\_ackがp\_max\_seqよりも大きかったらエラー制御処理を終了し、pipeの値を0に初期化し、snd\_cwndをssthreshに記憶していた値に設定し、snd\_unaとp\_max\_seqの値をti\_ackの値に設定してCongestion Avoidanceを開始する。

#### 【0023】

次に、図15と図16を用いて既存の技術の動作を説明する。図15、図16は送信装置1におけるデータセグメント送信時の処理とACKメッセージ受信時の処理の流れを示したものである。ただし、受信装置2から通知される受信ウィンドウサイズを用いたフロー制御や再送制御タイマのタイムアウト値計算、輻輳ウィンドウの更新等については本発明に直接関係が無いため記述を省略している。

#### 【0024】

データセグメント送信処理手段12は、エラー制御処理中で無いSlow Start、Congestion Avoidanceの場合(図15中、ステップS40でNo)には、輻輳制御とフロー制御の範囲内でデータセグメントを送信する(同、ステップS41～S44)。以降、転送されるデータセグメントやACKメッセージが損失せず、順序も変わらない場合、データセグメント受信処理手段23とACK送信処理手段22はデータセグメントを受信してその送達確認のためのACKメッセージを作成し、送信する。ACK受信処理手段13はACKメッセージを受信すると、この場合送信したデータセグメントに対して順序正しく累積確認が帰ってくるので(図16中、ステップS1でNo、S3でNo、S10でNo)、累積確認により送達が確認されたデータを送信バッファから削除し、この累積確認の内容を反映するようにスコアボード(scrb)を編集し(スコアボードに累積確認で送達確認されたリスト要素がある場合はそれを削除する

。先頭のリスト要素のretranの値がti\_ackよりも小さい場合はretranの値をti\_ackに設定)、snd\_cwndの値を更新し、snd\_unaとp\_max\_seqの値をこのACKメッセージで累積確認された値ti\_ackに設定してデータセグメント送信処理手段12に送信要求を行なう(同、ステップS20からS23)。

## 【 0 0 2 5 】

データセグメント送信処理手段12が送信したデータセグメントが損失し、それ以外の送信したデータセグメントが送信順通りにデータセグメント受信処理手段23に受信された場合、データセグメント受信処理手段23では損失したデータセグメント以外のデータセグメントを受信し、受信したデータセグメントを通知されたACK送信処理手段22は、損失したデータセグメント以降のシーケンス番号を持つデータセグメントの受信に対してはそのデータセグメントの受信を知らせるSACKオプションを含む重複ACKを作成し、その重複ACKを送信装置1に送信する。ACK受信処理手段13では、SACKオプションを含む重複ACKを受信するとまず、SACKオプション処理手段131がSACKオプションで通知されている情報をスコアボードに反映し(送達が通知されているデータセグメントをスコアボードに追加等)(同、ステップS1でYes、S2)、その重複ACKをACK処理手段132に渡す。ACK処理手段132では重複ACKを渡されると、重複ACK受信回数dupacksをカウントアップする(同、ステップS3でNo、S10でYes、S11実行、S12でNo)。重複ACKの受信回数が規定数rxmtthreshに達した場合(同、ステップS12でYes)は、ACK処理手段132は、その重複ACKで累積確認された次以降のデータセグメントを1つ再送し、エラー制御処理のための各種変数の初期設定を行なう(同、ステップS13、S14)。エラー制御処理のための変数の初期設定では、P\_max\_seqにsnd\_maxを記憶し、推定した滞留データサイズ $snd\_nxt - snd\_una - dupacks * segsz$ をpipeに設定し、ssthreshと輻輳ウィンドウサイズsnd\_cwndにそれまでの輻輳ウィンドウサイズの半分の値 $snd\_cwnd / 2$ を設定する。ここで、segszとは、MSSのことである。

## 【 0 0 2 6 】

以降、データセグメント送信処理手段12とACK受信処理手段13はp\_max\_seqで記憶しているデータセグメントが累積確認されるまでエラー制御処理を行なう。

## 【 0 0 2 7 】

エラー制御処理では、ACK処理手段132は、受信されたACKメッセージに対して、そのACKメッセージの累積確認がそれまで累積確認していたデータセグメントを更新しない場合（同、ステップS3とS30でそれぞれYes、S31でNo）はpipeの値をsegsz分小さくし、データセグメント送信処理手段12に送信要求をする（同、ステップS37、S36）。それまでの累積確認を更新するACKメッセージを受信した場合（同、ステップS3とS30とS31でYes）は、累積確認されたデータセグメントを送信バッファから削除し、この累積確認の内容をスコアボードに反映させ、pipeの値をsegsz分小さくし、snd\_unaの値をti\_ackに設定して、データセグメント送信処理手段12に送信要求をする（同、ステップS32～S36）。

## 【 0 0 2 8 】

エラー制御中に送信要求をされたデータセグメント送信処理手段12は、輻輳ウィンドウpipeの値がsnd\_cwndの値よりも小さい場合にデータセグメントを作成し、送信する（図15中、ステップS40とS50でYes）。その際にはデータセグメント送信処理手段12は、まず、スコアボードから再送すべきデータの有無を調べ、再送すべきデータがあればそのデータセグメントを作成し（同、ステップS51でYes、S52）、再送すべきデータが無く（同、ステップS51でNo）、送信すべきデータがあれば（同、ステップS53でYes）後続の新規のデータセグメントを作成する（同、ステップS54）。再送すべきデータセグメントとはこの場合スコアボードで送達が確認されていないデータセグメントで、かつ、一度も再送されていないデータセグメントのことを指し、このようなデータセグメントをシーケンス番号が小さい順に選択する。すなわち、スコアボードの先頭のリスト要素から、start > retran+1となるリスト要素を探し、そのようなリスト要素が存在すれば、そのリスト要素のretran+1からstart-1までが再送の対象となるデータセグメントとなる。

## 【 0 0 2 9 】

再送データセグメントの場合も、新規データセグメントの場合もいずれにおいてもpipeの値をそのデータセグメントのサイズ分増やし、そのデータセグメントを送信する（同、ステップS55、S56）。

## 【 0 0 3 0 】

一方、`p_max_seq`に記憶しているシーケンス番号のデータセグメントが累積確認された場合はデータセグメント送信処理手段12とACK受信処理手段13はエラー制御を終了するが、そのようなACKメッセージを受信した場合（図16中、ステップS3でYes、ステップS30でNo）、ACK処理手段132は、通常のACK処理と同様送達確認されたセグメントを送信バッファから削除し、累積確認の内容をスコアボードに反映させ、エラー制御のための各種変数を更新し、データセグメント送信処理手段12に送信要求を行なう（同、ステップS20～S23）。

### 【0031】

#### 【発明が解決しようとする課題】

従来の技術の第1の問題点は、再送されたデータセグメントが損失した場合にスループット特性が劣化することである。その理由は、従来の技術においては再送されたデータセグメントが損失した場合について何ら考慮されておらず、その損失を検出するには再送タイマのタイムアウトによる検出しか手段が無いためである。図17には第1の問題点により発生するデータセグメントとACKメッセージのシーケンス図の一例を示す。ここでは、送信側TCPと受信側TCPに張られたとある1コネクションでやり取りされるデータセグメントとACKメッセージのシーケンスを示している。データセグメントのシーケンス番号は1から始まっているが、これはTCPでのシーケンス番号の絶対値やコネクション開設時点からの相対値のいずれを表すものでもなく、説明を簡単にするためにここで取り上げるシーケンスの最初のデータセグメントのシーケンス番号を1にしたものである（以降のシーケンス図でも同様）。図17では輻輳ウィンドウが $8 * MSS$ で1つのデータセグメント（1：1000）が損失し、FRetで再送した同一データを含むデータセグメントが再度損失した場合のシーケンスについて示したものである。この場合、再送したデータセグメント1：1000が損失したためそのデータセグメントを送信したときに設定した再送タイマがタイムアウトするまでデータセグメントの損失が検出されない。

### 【0032】

なお、図17および他のシーケンス図において、“ $x : y$ ”（ $x, y$ は整数）はデータセグメント $x$ から $y$ を含むTCPデータセグメントを表す。“ $ACK : n$ ”（ $n$ は整数）

はデータセグメントnの累積確認を含むACKメッセージを表す。“SACK: x~y”は、SACKオプションによるデータセグメントxからyの受信通知を含むメッセージを表す。“X”印はパケット損失を表す。

## 【 0 0 3 3 】

従来の技術の第2の問題点は、SACKオプションでは、エラー制御処理中に送信されたデータセグメントの到着順序が入れ替わった場合、または、送信ウィンドウサイズが十分大きくない場合、スループット特性が劣化する場合があることである。その理由は、送信側TCPは受信側TCPからSACKオプションで通知される受信セグメントの情報から受信データの抜けを知ることができるが、送信側TCPが抜けを知った直後にそのデータのデータセグメントを再送するならば、データセグメントの到着順序が入れ替わったときに不要な再送を行なうからである。たとえば、送信側TCPが同じデータセグメントの抜けを通知するACKを規定数受信した後、そのデータセグメントを再送するようにしても、送信ウィンドウが規定数分のデータセグメントが送信されないような小さなサイズの場合は再送タイマのタイムアウトでしかデータセグメントの損失を検出できないからである。

## 【 0 0 3 4 】

従来の技術の第3の問題点は、エラー制御処理中に受信側TCPからのACKメッセージが損失した場合、スループット特性が劣化する場合があることである。その理由は、従来の技術ではエラー制御処理中の送信側TCPは受信側TCPからのACKメッセージ受信に対応して塞がった送信ウィンドウを開放するため、ACKメッセージが損失した場合はそれだけ塞がった送信ウィンドウが解放されないままになるためである。

## 【 0 0 3 5 】

図18には第3の問題点により発生するデータセグメントとACKメッセージのシーケンス図の一例を示す。図18では輻輳ウィンドウが8\*MSSで1つのデータセグメント(1:1000)が損失し、FRetで再送した同一データを含むデータセグメントが再度損失し、かつ、その後受信側TCPが送信したACKメッセージの内の一つが損失した場合のシーケンスについて示したものである。この場合、データセグメント1:1000の送達を確認されるまでは最初の輻輳ウィンドウサイズ8\*MSSの半

分の値、すなわち、 $4 * MSS$ 分を送信ウィンドウとして利用できるはずであるが、ACKメッセージの一つが損失したことにより損失後は実質 $3 * MSS$ 分しか送信ウィンドウとして利用していない。

## 【 0 0 3 6 】

第1の問題点に関しては、近藤、渥美、吉田、“タイムアウトを回避するSACK-TCP再送制御方式”、電子情報通信学会信学技報CQ98-63 pp. 21-26、1998年12月にこの問題点を解決する目的で、再送データセグメントの次に送信したデータセグメントのシーケンス番号を保存しておき、次シーケンス番号以降のデータセグメントの送達でSACKオプションによって通知された場合に再送データセグメントが損失と判断し、直ちに再送することが記載されているが、この技術には第2の問題点が存在する。

## 【 0 0 3 7 】

本発明の目的は、このような問題点を解決し、例えば引用文献Bで推奨されている輻輳制御の一般原則の制約の範囲内でスループット特性の良いデータ通信システム、その通信方法及びその通信プログラムを記録した記録媒体を提供することにある。

## 【 0 0 3 8 】

## 【課題を解決するための手段】

上記課題を解決するため、本発明の第1の態様は、データ送信装置とデータ受信装置がネットワークで接続され、前記データ送信装置は前記データ受信装置へデータパケットを送信し、前記データ受信装置は前記データ送信装置に受信したデータパケットの送達確認を通知するACKパケットを送信し、前記データ送信装置は前記ACKパケットの内容から送信したデータパケットの損失を検出し、該損失を検出したデータパケットを再送することによりデータパケットのエラー制御を行なう際に、前記データ送信装置は、送信するデータパケットのラウンドトリップ数を計数するカウンタを保持し、送信した各データパケットに対して、該データパケットと該データパケット送信時の前記カウンタの値の対応を記憶し、前記記憶したカウンタの値がその時点での前記カウンタの値より2つ以上小さい場合に、該記憶したカウンタ値に対応するデータパケットが損失したとみなし、該

データパケットを再送することを特徴とする。

【0039】

本発明の第2の態様は、第1の態様において、前記データ送信装置は、送信するデータパケットにその時点での前記カウンタ値をラウンドトリップ通知情報として含ませ、前記データ受信装置は、受信したデータパケットに対するACKパケットに該受信したデータパケットがラウンドトリップ通知情報として含む前記カウンタ値をラウンドトリップ応答情報として含ませ、前記データ送信装置は、受信したACKパケットにラウンドトリップ応答情報として含まれている値が前記カウンタの値と等しい場合に、前記カウンタの値を1つ増加させることを特徴とする。

【0040】

本発明の第3の態様は、データ送信装置とデータ受信装置がネットワークで接続され、前記データ送信装置は前記データ受信装置へデータパケットを送信し、前記データ受信装置は前記データ送信装置に受信したデータパケットの送達確認を通知するACKパケットを送信し、前記データ送信装置は前記ACKパケットの内容から送信したデータパケットの損失を検出し、該損失を検出したデータパケットを再送することによりデータパケットのエラー制御を行なう際に、前記データ送信装置は、送信するデータパケットのラウンドトリップ数を計数するカウンタと所定のデータを記憶する第1及び第2のテーブルを保持し、送信した各データパケットに対して、該データパケットと該データパケット送信時の時刻の対応を前記第1のテーブルに記憶し、前記カウンタの取る各カウンタ値に対して、カウンタ値と前記カウンタがそのカウンタ値を取った後初めてデータパケットを送信するときの送信時の時刻との対応を前記第2のテーブルに記憶し、前記第1のテーブルに記憶した時刻の値が、前記第2のテーブルに記憶したその時点での前記カウンタの値より2つ小さいカウンタ値に対応する時刻より早い場合に、該第1のテーブルに記憶した時刻に対応するデータパケットが損失したとみなし、該データパケットを再送することを特徴とする。

【0041】

本発明の第4の態様は、第3の態様において、前記データ送信装置は、送信す

るデータパケットにその時点での時刻を時刻通知情報として含ませ、前記データ受信装置は、受信したデータパケットに対するACKパケットに該受信したデータパケットが時刻通知情報として含む前記時刻を時刻応答情報として含ませ、前記データ送信装置は、受信したACKパケットに時刻応答情報として含まれている時刻が、前記第2のテーブルに記憶したその時点での前記カウンタの値と対応させて記憶させた時刻と等しいかまたは大きい場合に、前記カウンタの値を1つ増加させることを特徴とする。

#### 【0042】

本発明の第5の態様は、データ送信装置とデータ受信装置がネットワークで接続され、前記データ送信装置は前記データ受信装置へデータパケットを送信し、前記データ受信装置は前記データ送信装置に受信したデータパケットの送達確認を通知するACKパケットを送信し、前記データ送信装置は前記ACKパケットの内容から送信したデータパケットの損失を検出し、該損失を検出したデータパケットを再送することによりデータパケットのエラー制御を行ない、更には、データパケットの損失を検出後、該データパケットの送達がデータ送信装置によって確認されるまでのエラー制御中に送信ウィンドウにより送信フロー制御を行なう際に、前記データ送信装置は、エラー制御中に前記データ受信装置から受信したACKパケットで新規に送達確認がなされたデータの合計サイズ分の閉じられたウィンドウを前記送信ウィンドウから解放し、その分を送信可能とすることを特徴とする。

#### 【0043】

##### 【発明の実施の形態】

以下、本発明の実施の形態について図面を参照して詳細に説明する。図1には、本発明のデータ通信システム（データ通信装置）、その通信方法及びその通信プログラムを記録した記録媒体に係る第1の実施形態が示されている。

#### 【0044】

図1に示されるように、本発明の第1の実施の形態は、送信装置1と受信装置2とネットワーク3から構成され、送信装置1と受信装置2はネットワーク3を介して接続される。送信装置1は、アプリケーションサービス実行手段11とデータセグ

メント送信処理手段121、ACK受信処理手段13、パケット送信手段14、パケット受信手段15から構成される。受信装置2は、アプリケーションサービス実行手段21とACK送信処理手段221、データセグメント受信処理手段231、パケット送信手段24、パケット受信手段25から構成される。アプリケーションサービス実行手段11とアプリケーションサービス実行手段21は、既存の技術で説明したものと同一のもので、対になって利用者にサービスを提供する手段である。パケット送信手段14、24とパケット受信手段15、25は、既存の技術で説明したものと同一でそれぞれ上位に対してパケット送信・受信のサービスを提供するものである。ネットワーク3も既存の技術で説明したものと同一でIPネットワークである。

## 【 0 0 4 5 】

データセグメント送信処理手段121とACK受信手段13、ACK送信手段221、データセグメント受信処理手段231も、既存の技術で説明したものと同様、アプリケーションサービス実行手段11から渡されるデータをアプリケーション実行手段21に誤りなく渡すためのプロトコル処理を行なう。

## 【 0 0 4 6 】

データセグメント送信処理手段121とACK受信処理手段13は送信したデータセグメントのラウンドトリップの回数を記憶するラウンドトリップカウンタ（RTC）を持ち（図2の「ラウンドトリップカウンタ」）、既存の技術で説明したスコアボードの代わりに送信セグメントリスト（SSLとも記述する）を持つ。送信セグメントリストはこれまでデータセグメント送信処理手段121が送信したデータセグメントの情報を格納するリスト要素のリストで構成されている。各リスト要素にはこれまで送信して累積確認とSACKオプションで送達が確認されていないデータセグメントの最初と最後のデータのシーケンス番号と、そのデータセグメントが最後に送信されたときのRTCの値が格納されており（同、「送信セグメントリスト」の「start」、「end」、「rt」）、これらのリスト要素がそれぞれ最後に送信された順に並んでいる。例えば、図2で示した送信セグメントリストでは、データセグメント1：1000をRTCが1のときに送信し、データセグメント1001：2000と2001：3000をRTCが2のとき送信し、これら3つのデータセグメントに対する累積確認、または、SACKオプションによる送達確認はまだ受信されていないときの

送信セグメントリストを表している。ここで、FRetなどによりデータセグメント 1 : 1000 を RTC が 3 のときに再送した場合、送信セグメントリストは、先頭からデータセグメント 1001 : 2000、2001 : 3000、1 : 1000 のリスト要素の並びになり、かつ、データセグメント 1001 : 2000 と 2001 : 3000 のリスト要素の内容は図 2 のそれぞれのリスト要素と同一のものになっているが、データセグメント 1 : 1000 のリスト要素の rt の値が 3 となったものになる。

## 【 0 0 4 7 】

また、送信装置 1 と受信装置 2 で送受信されるメッセージには図 3 で示すラウンドトリップカウンタ値を通知する Round Trip Counter (RTC) オプションフィールドと Round Trip Counter Echo (RTCE : ラウンドトリップカウンタ・エコー ; ラウンドトリップ応答) オプションフィールドが追加する。すなわち、データセグメント送信処理手段 121 はデータセグメントを送信する際、作成したデータセグメントに RTC オプションフィールドを追加し、そのとき保持している RTC の値を RTC オプションフィールドの Round Trip Counter 値フィールド (以降、ti\_rtc フィールドとも表記する) に格納してデータセグメントを送信する。図 3 に示す例では、RTC オプションフィールドはデータの種別 (「RTC」) を示す 8 ビットの Kind (カインド) フィールドと、8 ビットを単位としてデータの長さ 3 を表す 8 ビットの Length (レングス) フィールドと、RTC 値を示す 8 ビットの RTC 値フィールドから構成されている。RTCE オプションフィールドはデータの種別 (「RTCE」) を示す 8 ビットの Kind フィールドと、データの長さ 3 を表す 8 ビットの Length フィールドと、RTCE 値を示す 8 ビットの RTCE 値フィールドから構成されている。また、ACK 送信処理手段 221 は、データセグメント受信処理手段 231 から通知される受信したデータセグメントの ACK メッセージを作成する際、作成した ACK メッセージに RTCE オプションフィールドを追加し、データセグメントの RTC オプションの Round Trip Counter 値フィールドの値を Round Trip Counter Echo 値フィールド (以降、ti\_rtce フィールドとも表記する) に格納して ACK メッセージを送信する。

## 【 0 0 4 8 】

データセグメント送信処理手段 121 は、アプリケーションサービス実行手段 11 から渡されるデータを送信バッファに蓄積し、送信バッファに蓄積されているデ

ータからTCPセグメントを生成してパケット送信手段14に渡すことを基本機能とし、さらに、本発明の第一の実施の形態における輻輳制御とフロー制御の一端とSACKオプションを利用した場合のエラー回復制御の一端を備える。

## 【 0 0 4 9 】

データセグメント送信処理手段121が備える本発明の第1の実施の形態における輻輳制御とフロー制御の一端とは、輻輳制御とフロー制御におけるTCPセグメント送信の可否を判断することである。既存の技術で示したとおり、本発明の第1の実施の形態においても(1)累積確認されたデータの1バイト後続のデータのシーケンス番号を指すsnd\_una、(2)次に送信すべきデータの先頭のデータのシーケンス番号を指すsnd\_nxt、(3)相手側からTCPヘッダのWindowフィールドで通知された相手側の受信ウィンドウサイズを指すsnd\_wnd、輻輳制御のための送信ウィンドウである輻輳ウィンドウsnd\_cwndの各情報を管理している。データセグメント送信処理手段121は、送達確認が取れていない送信済みデータの合計サイズが相手側受信ウィンドウsnd\_wndと輻輳ウィンドウsnd\_cwndのいずれよりも小さい場合にのみ後続のデータからデータセグメントを生成し、送信する。また、その際には作成するデータセグメントにRTCオプションを付加し、そのRound Trip Counter値フィールドにそのとき保持しているRTCの値を格納し、かつ、作成したデータセグメントに関するリスト要素(startとendの値がデータセグメントの最初と最後のデータのシーケンス番号であり、rtの値がRound Trip Counter値フィールドに格納したRTCの値)を作成し、リスト要素を送信セグメントリストの末尾に追加する。

## 【 0 0 5 0 】

データセグメント送信処理手段121が備えるSACKオプションを利用したエラー回復制御の一端とは、エラー回復処理中のデータセグメント送信の可否の判断と再送データセグメントの選定、エラー回復処理中のネットワークに滞留している転送中のデータの推定値(以降、pipeと記述する)の値の管理である。データセグメント送信処理手段121は、既存の技術と同様、エラー回復処理中には滞留データ推定値pipeが輻輳ウィンドウsnd\_cwndを下回るときのみその差分を最大サイズとしてデータを送信する。データを送信する際は、累積確認とSACKオプション

で送達確認が通知されていないデータの内、リスト要素のrtの値がRTC-2と同じかそれより小さい値を持つリスト要素のデータを送信セグメントリストの先頭方向から順に選択して再送する。再送するデータが存在しない場合には未送信の後続のデータを送信する。いずれにおいても、データセグメント送信後は送信したデータセグメントのサイズを滞留データ推定値pipeに加算する。

## 【 0 0 5 1 】

また、SACKオプションで通知されたデータセグメント送達情報は既存の技術で説明したスコアボードでは無く、送信セグメントリストと呼ばれるリストで管理している。送信セグメントリストの構造は前述のとおりである。

## 【 0 0 5 2 】

データセグメント受信処理手段231は、既存の技術で説明したデータセグメント受信処理手段23と同様、パケット受信手段25から渡されるデータセグメント送信処理手段121から送信されたデータセグメントを受信し、受信したデータを順序正しく受信バッファに格納する。また、受信したデータセグメントの情報をACK送信処理手段221に通知する。受信したデータセグメントにRTCオプションが付加されている場合はその情報もACK送信処理手段221に渡す。

## 【 0 0 5 3 】

ACK送信処理手段221はデータセグメント受信処理手段231から通知されるデータセグメントの情報からACKメッセージを作成し、パケット送信手段24に渡す。ACK送信処理手段221は、これまで受信したデータセグメントが含むデータに抜けがあり、幾つかのデータブロックが存在する場合、既存の技術同様SACKオプションを用いてデータブロックを通知する。また、データセグメント受信処理手段231から渡されたデータセグメントの情報にRTCオプションがある場合は作成したACKメッセージにRTCEオプションを付加し、そのRound Trip Counter Echo値フィールドに通知されたRTCオプションのRound Trip Counter値フィールドの値を格納する。

## 【 0 0 5 4 】

ACK受信処理手段13はパケット受信手段15から渡されるACK送信手段221が送信したACKメッセージを処理する手段で、更に、SACKオプション処理手段133とACK

処理手段134から構成される。SACKオプション処理手段133は、パケット受信手段15から渡されたACKメッセージにSACKオプションが付いている場合、その内容を送信セグメントリストに反映させる。具体的には、リスト要素が含むデータセグメントがSACKオプションで通知された送達確認されたデータに含まれるようなリスト要素を送信セグメントリストから削除する。更に、エラー制御処理中の場合は、pipeの値を削除したリスト要素が含んでいたデータのサイズ分だけ減らす。

## 【 0 0 5 5 】

ACK処理手段134は、まず、受信したACKメッセージがRTCEオプションフィールドを含んでいる場合、RTCEオプションフィールドのRound Trip Counter Echo値フィールドti\_rtceの値がそのときのRTCの値と等しければRTCの値を一つ分カウントアップする。そして、エラー制御処理中でない場合、すなわち、Slow StartやCongestion Avoidance中は重複ACKを監視し、重複ACKが規定回数rxmtthresh回連続で受信された場合は、FRetの処理を行なう。その際には、既存の技術同様、重複ACKで累積確認されたデータの次のデータを再送するが、この場合は既存の技術と異なり、送信するデータセグメントにRTCオプションを付加し、送信するデータのデータセグメントに関する情報を持つリスト要素を送信セグメントリストから削除し、送信するデータのデータセグメントの情報を持つリスト要素（“start”と“end”がそれぞれそのデータセグメントが含む最初と最後のデータのシーケンス番号の値を持ち、“rt”がその時点でのRTCの値を持つリスト要素）を送信セグメントリストの末尾に追加する。そして、エラー制御処理中の場合の処理のため初期化処理として、その時点で送信済みであるデータの最後のデータのセグメント番号snd\_maxをp\_max\_seqに記憶し、推定した滞留データサイズをpipeに設定し、ssthreshと輻輳ウィンドウsnd\_cwndにその時点での輻輳ウィンドウの半分の値を記憶する。

## 【 0 0 5 6 】

ACK処理手段134は、エラー制御処理中の場合、（1）重複ACKに対してはpipeの値を1MSSサイズ分小さくし、（2）累積確認が更新された場合には、累積確認されたデータを送信バッファから削除し、その内容を送信セグメントリストに反映し（累積確認されたデータの情報を格納するリスト要素を送信セグメントリスト

から削除)、削除したリストが含んでいたデータのサイズの合計をpipeの元の値から減じたものをpipeの値として再設定し、snd\_unaをACKメッセージの通知する累積確認の値ti\_ackに更新する。その後、上記、(1)、(2)いずれの場合もデータセグメント送信処理手段11に送信要求を行なう。FRet起動時に送信していたデータセグメントが全て送達確認されたら、すなわち、ti\_ackがp\_max\_seqよりも大きかったらエラー制御処理を終了し、pipeの値を0に初期化し、snd\_cwndをsthreshに記憶していた値に設定し、snd\_unaとp\_max\_seqの値をti\_ackの値に設定してCongestion Avoidanceを開始する。

## 【 0 0 5 7 】

次に、図4と図5を参照して本発明の第1の実施の形態における動作を説明する。図4と図5は送信装置1におけるデータセグメント送信時の処理とACKメッセージ受信時の処理の流れを示したものである。ただし、受信装置2から通知される受信ウィンドウサイズを用いたフロー制御や再送制御タイマのタイムアウト値計算、輻輳ウィンドウの更新等については本発明に直接関係が無いと見做すため記述を省略している。

## 【 0 0 5 8 】

データセグメント送信処理手段121は、エラー制御処理中でないSlow Start、Congestion Avoidanceの場合(図4中、ステップS40でNo)には、輻輳制御とフロー制御の範囲内でデータセグメントSを送信し、その際にはデータセグメントにRTCオプションを付加し、送信セグメントリストにデータセグメントに関する情報を格納し(同、ステップS41～S43およびX10～X11)、データセグメントを送信する(同、ステップS44)。以降、転送されるデータセグメントやACKメッセージが損失せず、順序も変わらない場合、データセグメント受信処理手段231とACK送信処理手段221はデータセグメントを受信してその送達確認のためのACKメッセージを作成し、送信する。この際、ACK送信処理手段221は、各ACKメッセージにRTCEオプションフィールドを付加し、そのRTCEオプションフィールドのRound Trip Counter Echo値フィールドには、そのACKメッセージを送信する契機となったデータセグメントに付加されているRTCオプションフィールドのRound Trip Counter値フィールドの値を格納する。ACK受信処理手段13はACKメッセージを受信すると

、この場合、SACKオプションが付加されていないためSACKオプション処理手段133の処理は行なわず（図5中、ステップS1でNo）、受信したACKメッセージはACK処理手段134に渡される。ACK処理手段134は、受信したACKメッセージがRTCEオプションフィールドを含んでいる場合、まず、RTCEオプションフィールドのRound Trip Counter Echo値フィールドti\_rtceの値を参照し、ti\_rtceの値がそのときのRTCの値と等しければ（同、ステップX4-1でYes）RTCの値を一つ分カウントアップする（同、ステップX4-2）。そして、ACK処理手段134は、この場合送信したデータセグメントに対して順序正しく累積確認が帰ってくるので（同、ステップS3とS10でNo）、累積確認により送達が確認されたデータを送信バッファから削除し、この累積確認の内容を反映するように送信セグメントリストを編集し（送信セグメントリストに累積確認で送達確認されたリスト要素がある場合はそれを削除）、snd\_cwndの値を更新し、snd\_unaとp\_max\_seqの値をこのACKメッセージで累積確認された値ti\_ackに設定してデータセグメント送信処理手段121に送信要求を行なう（同、ステップS20、X5、S22からS23）。

#### 【 0 0 5 9 】

データセグメント送信処理手段121が送信したデータセグメントが損失し、それ以外の送信したデータセグメントが送信順通りにデータセグメント受信処理手段231に受信された場合、データセグメント受信処理手段231では損失したデータセグメント以外のデータセグメントを受信し、受信したデータセグメントを通知されたACK送信処理手段221は、損失したデータセグメント以降のシーケンス番号を持つデータセグメントの受信に対してはそのデータセグメントの受信を知らせるSACKオプションを含む重複ACKを作成し、その重複ACKを送信装置1に送信する。ACK受信処理手段13では、SACKオプションを含む重複ACKを受信するとまず、SACKオプション処理手段133がSACKオプションで通知されている情報を送信セグメントリストに反映し（送達が通知されているデータセグメントのリスト要素を送信セグメントリストから削除）（図5中、ステップS1でYes、X1）、重複ACKをACK処理手段134に渡す。ACK処理手段134では重複ACKを渡されると、重複ACK受信回数カウンタdupacksをカウントアップする（同、ステップS3でNo、S10でYes、S11実行、S12でNo）。重複ACKの受信回数が規定数rxmtthreshに達した場合（同、ス

テップS12でYes) は、ACK処理手段134は、その重複ACKで累積確認された次以降のデータセグメントを1つ再送し、データセグメント再送を反映するように送信セグメントリストを編集し(同、ステップX9)、エラー制御処理のための各種変数の初期設定を行なう(同、ステップS14)。エラー制御処理のための変数の初期設定では、`p_max_seq`に`snd_max`を記憶し、推定した滞留データサイズ`snd_nxt - snd_una - dupacks * segsz`を`pipe`に設定し、`ssthresh`と輻輳ウィンドウサイズ`snd_cwnd`にそれまでの輻輳ウィンドウサイズの半分の値`snd_cwnd/2`を設定する。ここで、`segsz`とは、MSSのことである。

## 【 0 0 6 0 】

以降、データセグメント送信処理手段121とACK受信処理手段13は`p_max_seq`で記憶しているデータセグメントが累積確認されるまでエラー制御処理を行なう。

## 【 0 0 6 1 】

エラー制御処理では、ACK処理手段134は、受信されたACKメッセージに対して、そのACKメッセージの累積確認がそれまで累積確認していたデータセグメントを更新しない場合(同、ステップS3とS30でそれぞれYes、S31でNo)は`pipe`の値を`segsz`分小さくし、データセグメント送信処理手段121に送信要求をする(同、ステップS37、S36)。それまでの累積確認を更新するACKメッセージを受信した場合(同、ステップS3とS30とS31でYes)は、累積確認されたデータセグメントを送信バッファから削除し(同、ステップS32)、この累積確認の内容を送信セグメントリストに反映させ(リスト要素が含むデータセグメントが累積確認で送達確認されたデータに含まれるリスト要素を送信セグメントリストから削除)(同、ステップX6)、`pipe`の値を前述のステップX6で削除したリスト要素が含んでいたデータのサイズの合計分減じ(同、ステップX7)、`snd_una`の値を`ti_ack`に設定して(同、ステップS35)、データセグメント送信処理手段12に送信要求をする(同、ステップS36)。

## 【 0 0 6 2 】

エラー制御中に送信要求をされたデータセグメント送信処理手段121は、輻輳ウィンドウ`pipe`の値が`snd_cwnd`の値よりも小さい場合にデータセグメントを作成し、送信する(図4中、ステップS40とS50でYes)。その際にはデータセグメント

送信処理手段121は、まず、送信セグメントリストから再送すべきデータの有無を調べ、再送すべきデータがあれば（同、ステップX12でYes）、そのデータセグメントを作成し（同、ステップS52）、生成したデータセグメントに対応するリスト要素を送信セグメントリストから削除し（同、ステップX13）、pipeの値を前記ステップX13で削除したリスト要素が含むデータのサイズ分、すなわち、生成したデータセグメントのサイズ分、減じる（同、ステップX14）。再送すべきデータが無く（同、ステップX12でNo）、送信すべきデータがあれば（同、ステップS53でYes）後続の新規のデータセグメントを作成する（同、ステップS54）。再送すべきデータとはこの場合送信セグメントリストのリスト要素が記憶しているデータで、かつ、リスト要素のrtの値が現在のラウンドトリップカウンタRTCの値より2以上小さい値になっているリスト要素が記憶しているデータのことであり、データセグメント送信処理手段121はこのようなデータを送信セグメントリストの先頭方向から順に選択する。すなわち、データセグメント送信処理手段121は、送信セグメントリストの先頭のリスト要素から、 $rt \leq RTC - 2$ となるリスト要素を探し、そのようなリスト要素が存在すれば、そのリスト要素のstartからendまでを再送の対象とする。

## 【 0 0 6 3 】

再送データセグメントの場合も、新規データセグメントの場合もいずれにおいてもpipeの値をそのデータセグメントのサイズ分増やし（同、ステップS55）、そのデータセグメントにRTCオプションフィールドを付加してそのRound Trip Counter値フィールドti\_rtcに現在のラウンドトリップカウンタRTCの値を設定し（同、ステップX15）、データセグメント送信の内容を示すリスト要素（startとendの値がデータセグメントが含むデータの先頭と末尾のデータのシーケンス番号で、rtの値がデータセグメントのti\_rtcの値）を作成してリスト要素を送信セグメントリストの末尾に追加し（同、ステップX16）、そのデータセグメントを送信する（同、ステップS56）。

## 【 0 0 6 4 】

p\_max\_seqに記憶しているシーケンス番号のデータセグメントが累積確認された場合はデータセグメント送信処理手段121とACK受信処理手段13はエラー制御を

終了するが、そのようなACKメッセージを受信した場合（図5中、ステップS3でYes、S30でNo）、ACK処理手段134は、通常のACK処理と同様送達確認されたセグメントを送信バッファから削除し（同、ステップS20）、累積確認されたデータの内容を持つリスト要素を送信セグメントリストから削除し（同、ステップX5）、エラー制御のための各種変数を更新し（同、ステップS22）、データセグメント送信処理手段121に送信要求を行なう（同、ステップS23）。

#### 【 0 0 6 5 】

以上のように、ラウンドトリップカウンタを導入し、送信するデータセグメントにラウンドトリップカウンタ値を含め、送信したデータセグメントと送信時のラウンドトリップカウンタ値を関連させて記憶しておき、ACKメッセージに含まれるRound Trip Counter Echo値フィールドの値を利用することにより再送したデータセグメントの損失を検出し、再度、再送することができ、スループット特性を向上させることができる。例えば、既存の技術の課題で説明した図17のシーケンスと同様な状況においては、図6で示す通り、再送したデータセグメント1:1000の損失を後続のデータセグメントでやり取りされるRTCの値で検出し、再送タイマのタイムアウトを待たずデータセグメント1:1000の再度の再送を行なうことができる（図6ではrtc=4で再度の再送を行っている。）。

#### 【 0 0 6 6 】

また、送信セグメントリストからリスト要素を削除する際に、削除したリスト要素が示すデータセグメントのサイズをpipeから減じることにより、エラー制御処理中のACKメッセージの損失に対して、それにより1MSS分輻輳ウィンドウ分を使い切れなような状況を回避し、スループット特性を向上させることができる。例えば、既存の技術の課題で説明した図18のシーケンスと同様な状況においては、一つのACKメッセージが損失した後では1MSS分の輻輳ウィンドウが使用されていない状況になったが、本発明の第1の実施の形態においては、図7で示す通り、ACKメッセージが損失した後も後続のACKメッセージ受信により1MSS分の輻輳ウィンドウが使用されないような状況は回避される。

#### 【 0 0 6 7 】

本第1の発明の実施の形態では、データの送信のみを行なうアプリケーション

サービス実行手段11を搭載する送信装置1とデータの受信のみを行なうアプリケーションサービス実行手段21を搭載する受信装置2からなる構成を採っているが、一つの装置にデータの送受信を行なうようなアプリケーションサービス実行手段を搭載し、かつ、データセグメント送信処理手段121とデータセグメント受信処理手段231、ACK送信処理手段221、ACK受信処理手段13、パケット送信手段14、パケット受信手段15を搭載させることにより、本発明の第1の実施の形態の効果をもち、データの送信、受信を行なえる装置を構成することができる。

## 【 0 0 6 8 】

また、その際に、図3に示した二つのオプションをLengthフィールドの値として4を持ち、Kindフィールドに新規のオプション識別子の値を持ち、Lengthフィールド以降にRound Trip Counter値フィールドと、Round Trip Counter Echo値フィールドを持つRound Trip Counterオプションフィールドを定義することにより、送信、受信の双方向においてRound Trip Counterのやり取りのためのオプションフィールドサイズを最適化することができる。

## 【 0 0 6 9 】

次に、図8から図12を参照して、本発明の第2の実施の形態について説明する。上述した第1の実施の形態においては、送信装置1にラウンドトリップカウンタRTCを導入し、送信装置1でのデータセグメント送信に対して送信時のRTCの値と関連付けて管理し、RTCの値はデータセグメントのRTCフィールドと受信装置2から返されるACKメッセージのRTCEフィールドで明示的にやり取りしていた。これに対して、第2の実施の形態では、第1の実施の形態と同様にラウンドトリップカウンタRTCを導入するが、RTCの値のやり取りを“TCP Extensions for High Performance”，RFC1323で既に規定されているTCPのタイムスタンプオプションを利用する形態をとる。

## 【 0 0 7 0 】

本発明の第2の実施の形態の構成を図8に示す。送信装置1、受信装置2、ネットワーク3の接続形態と、送信装置1のアプリケーションサービス実行手段11、パケット送信手段14、パケット受信手段15、受信装置2のアプリケーションサービス実行手段21、パケット送信手段24、パケット受信手段25は第1の実施の形態と

同様であるが、送信装置1のデータセグメント送信処理手段122、SACKオプション処理手段135、ACK処理手段136と、受信装置2のACK送信処理手段222とデータセグメント受信処理手段232が異なる。

#### 【0071】

データセグメント送信処理手段122とACK受信処理手段13では、第1の実施の形態と同じラウンドトリップカウンタRTCを用いるが、送信セグメントリストのリスト要素の構成が異なり、また、新たにラウンドトリップカウンタ値と時刻を記録するラウンドトリップテーブルが追加されている（図9）。第2の実施の形態における送信セグメントリストのリスト要素は、第1の実施の形態におけるリスト要素と同様、送信したデータセグメントに関する情報を記憶し、対応するデータセグメントの含むデータの最初と末尾のデータのシーケンス番号をそれぞれ“start”と“end”に格納するが、第1の実施の形態のリスト要素と異なるのはデータセグメントにはタイムオプションフィールド（図10）を付加し、リスト要素の“TS”に対応するデータセグメントに付加したタイムスタンプオプションのTime Stamp Value（タイムスタンプバリュー）フィールド（図10中、Time Stamp Valueフィールド（TSvalフィールド）。以降、ti\_tsvalフィールドと記述する）の値を記憶しておくことである。また、ラウンドトリップテーブルはラウンドトリップカウンタ（RTC）フィールドと時刻フィールドからなるレコードのテーブルで、各ラウンドトリップカウンタの値をRTCフィールドに、RTCフィールドのラウンドトリップカウンタ値に対して、そのラウンドトリップカウンタ値において最初に送信したデータセグメントに付加したタイムスタンプオプションのTSvalフィールド値ti\_tsvalを時刻フィールドに格納しておく。

#### 【0072】

データセグメント送信処理手段122は、第1の実施の形態のデータセグメント送信処理手段121同様、エラー制御処理中で無い場合には輻輳制御とフロー制御の制限内で送信すべきデータがある場合にデータセグメントを生成し、送信するが、その際には各データセグメントにタイムスタンプオプションを付加し、そのti\_tsvalフィールド中に現在の時刻を格納し、データセグメント送信に対する前述した第2の発明の実施の形態におけるリスト要素を作成し、作成したリスト要

素を送信セグメントリストの末尾に追加する。また、現在のラウンドトリップカウンタ値に対して、その値の時刻がラウンドトリップテーブルに記録されていない場合はデータセグメントのti\_tsvalフィールドに付加した値を時刻フィールドの値として記録する。

#### 【 0 0 7 3 】

また、データセグメント送信処理手段122は、エラー制御中も本発明の第1の実施の形態のデータセグメント送信処理手段121同様、 $\text{snd\_cwnd} \geq \text{pipe}$ の場合に $\text{pipe} - \text{snd\_cwnd}$ を最大サイズとするデータセグメントを再送または新規送信し、送信時にはpipeの値を更新するが、再送または新規送信するデータセグメントを生成する際にはタイムスタンプオプションをデータセグメントに付加し、データセグメント送信に対する前述した第2の実施の形態におけるリスト要素を作成し、作成したリスト要素を送信セグメントリストの末尾に追加する。更には、現在のラウンドトリップカウンタ値に対して、その値の時刻がラウンドトリップテーブルに記録されていない場合はデータセグメントのti\_tsvalフィールドに付加した値を時刻フィールドの値として記録する。

#### 【 0 0 7 4 】

データセグメント受信手段232は、第1の実施の形態で説明したデータセグメント受信処理手段231と同様、パケット受信手段25から渡されるデータセグメント送信処理手段122から送信されたデータセグメントを受信し、受信したデータを順序正しく受信バッファに格納し、受信したデータセグメントの情報をACK送信処理手段222に通知する。受信したデータセグメントにタイムスタンプオプションが付加されている場合はその情報もACK送信処理手段222に渡す。

#### 【 0 0 7 5 】

ACK送信処理手段222はデータセグメント受信処理手段232から通知されるデータセグメントの情報からACKメッセージを作成し、パケット送信手段24に渡す。ACK送信処理手段222は、これまで受信したデータセグメントが含むデータに抜けがあり、幾つかのデータブロックが存在する場合、既存の技術同様SACKオプションを用いてデータブロックを通知する。また、データセグメント受信処理手段232から渡されたデータセグメントの情報にタイムスタンプオプションがある場合

は作成したACKメッセージにタイムスタンプオプションを付加し、そのTime Stamp Echo Replyフィールド（図10、Time Stamp Echo Replyフィールド。以降、ti\_tsecrフィールドと記述する）に通知されたタイムスタンプオプションのti\_tsvalフィールドの値を格納する。

#### 【 0 0 7 6 】

データセグメント受信処理手段232とACK送信処理手段222の動作は、既存の技術におけるSACKオプションとタイムスタンプオプションに対応したTCPスタックの動作と同じである。

#### 【 0 0 7 7 】

ACK受信処理手段13はパケット受信手段15から渡されるACK送信手段222が送信したACKメッセージを処理する手段で、更に、SACKオプション処理手段135とACK処理手段136から構成される。SACKオプション処理手段135は、第1の実施の形態のSACKオプション処理手段133と同様、パケット受信手段15から渡されたACKメッセージにSACKオプションが付いている場合、その内容を送信セグメントリストに反映させる。具体的には、リスト要素が含むデータセグメントがSACKオプションで通知された送達確認されたデータに含まれるようなリスト要素を送信セグメントリストから削除する。更に、エラー制御処理中の場合は、pipeの値を削除したリスト要素が含んでいたデータのサイズ分だけ減らす。

#### 【 0 0 7 8 】

ACK処理手段136は、まず、受信したACKメッセージがタイムスタンプオプションフィールドを含んでいる場合、タイムスタンプオプションフィールドのti\_tsecrの値が、Round TripテーブルのそのときのRTCの値をRTCフィールドの値として持つレコードの時刻フィールドの値（RTTable [RTC] と表現する；すなわちRTTable [x] はラウンドトリップテーブルのRTCフィールドがxの時刻フィールドの値を表す）と同じか、それより大きければRTCの値を一つ分カウントアップする。そして、以降の処理は第1の実施の形態のACK処理手段134と同様、エラー制御処理中でない場合、すなわち、Slow StartやCongestion Avoidance中は重複ACKを監視し、重複ACKが規定回数rxmtthresh回連続で受信された場合は、FRetの処理を行なう。その際には、重複ACKで累積確認されたデータの次のデータを再送する

が、この場合は第 1 の実施の形態の ACK 処理手段 134 と異なり、送信するデータセグメントにタイムスタンプオプションを付加し、送信するデータのデータセグメントに関する情報を持つリスト要素を送信セグメントリストから削除し、送信するデータのデータセグメントの情報を持つリスト要素（“start” と “end” がそれぞれデータセグメントが含む最初と最後のデータのシーケンス番号の値を持ち、“TS” がデータセグメントに付加した ti\_tsval の値を持つリスト要素）を送信セグメントリストの末尾に追加し、ラウンドトリップテーブルにその時点での RTC の値に対する時刻が設定されていない場合は ti\_tsval の値を時刻フィールドに設定する。そして、第 1 の実施の形態の ACK 処理手段 134 と同様、各種変数の設定を行なう。ACK 処理手段 136 は、エラー制御処理中の場合、操作する送信セグメントリストが第 2 の実施の形態の送信セグメントリストに変更されている点を除いては、全て第 1 の実施の形態の ACK 処理手段 134 と同様に処理を行なう。

## 【 0 0 7 9 】

次に、図 11、図 12 を参照して、本発明の第 2 の実施の形態の動作について説明する。図 11 と図 12 は送信装置 1 におけるデータセグメント送信時の処理と ACK メッセージ受信時の処理の流れを示したものである。ただし、受信装置 2 から通知される受信ウィンドウサイズを用いたフロー制御や再送制御タイマのタイムアウト値計算、輻輳ウィンドウの更新等については本発明に直接関係が無いため記述を省略している。

## 【 0 0 8 0 】

データセグメント送信処理手段 122 は、エラー制御処理中で無い Slow Start、Congestion Avoidance の場合（図 11 中、ステップ S40 で No）には、輻輳制御とフロー制御の範囲内でデータセグメントを送信し、その際にはデータセグメントにタイムスタンプオプションを付加し、送信セグメントリストにデータセグメントに関する情報を格納する（同、ステップ S41 ～ S43 および Y10 ～ Y11）。そして、Round Trip テーブルに現在の RTC の値を RTC フィールドの値に持つレコードの時刻フィールドに時刻が設定されていない場合（同、ステップ Y12 で Yes）、レコードの時刻フィールドに前記データセグメントの ti\_tsval の値を設定し（同、ステップ Y13）、データセグメントを送信する（同、ステップ S44）。以降、転送されるデー

タセグメントやACKメッセージが損失せず、順序も変わらない場合、データセグメント受信処理手段232とACK送信処理手段222はデータセグメントを受信してその送達確認のためのACKメッセージを作成し、送信する。この際、ACK送信処理手段222は、各ACKメッセージにタイムスタンプオプションフィールドを付加し、そのタイムスタンプオプションフィールドのti\_tsecrフィールドには、そのACKメッセージを送信する契機となったデータセグメントに付加されているタイムスタンプオプションのti\_tsvalフィールドの値を格納する。ACK受信処理手段13はACKメッセージを受信すると、この場合、SACKオプションが付加されていないためSACKオプション処理手段135の処理は行なわず（図12中、ステップS1でNo）、受信したACKメッセージはACK処理手段136に渡される。

#### 【 0 0 8 1 】

ACK処理手段136は、受信したACKメッセージがタイムスタンプオプションフィールドを含んでいる場合、まず、タイムスタンプオプションフィールドのti\_tsecrフィールドの値を参照し、ti\_tsecrの値がRound TripテーブルのそのときのRTCの値をRTCフィールドの値として持つレコードの時刻フィールドの値RTTable [RTC] と等しいか、それよりも大きければ（同、ステップY2-1でYes）RTCの値を一つ分カウントアップする（同、ステップY2-2）。そして、ACK処理手段136は、この場合送信したデータセグメントに対して順序正しく累積確認が帰ってくるので（同、ステップS3とS10でNo）、累積確認により送達が確認されたデータを送信バッファから削除し（同、ステップS20）、この累積確認の内容を反映するように送信セグメントリストを編集し（送信セグメントリストに累積確認で送達確認されたリスト要素がある場合はそれを削除）（同、ステップY3）、以降、第1の実施の形態のACK処理手段134と同様に各種変数の値を設定し、データセグメント送信処理手段122に送信要求を行なう（同、ステップS22、S23）。

#### 【 0 0 8 2 】

データセグメント送信処理手段122が送信したデータセグメントが損失し、それ以外の送信したデータセグメントが送信順通りにデータセグメント受信処理手段232に受信された場合、データセグメント受信処理手段232では損失したデータセグメント以外のデータセグメントを受信し、受信したデータセグメントを通知

されたACK送信処理手段222は、損失したデータセグメント以降のシーケンス番号を持つデータセグメントの受信に対してはそのデータセグメントの受信を知らせるSACKオプションを含む重複ACKを作成し、その重複ACKを送信装置1に送信する。ACK受信処理手段13では、SACKオプションを含む重複ACKを受信するとまず、SACKオプション処理手段135がSACKオプションで通知されている情報を送信セグメントリストに反映し（送達が通知されているデータセグメントのリスト要素を送信セグメントリストから削除）（同、ステップS1でYes、Y1）、重複ACKをACK処理手段136に渡す。ACK処理手段136では重複ACKを渡されると、重複ACK受信回数カウンタdupacksをカウントアップする（同、ステップS3でNo、S10でYes、S11実行、S12でNo）。重複ACKの受信回数が規定数rxmtthreshに達した場合（同、ステップS12でYes）は、ACK処理手段136は、その重複ACKで累積確認された次以降のデータセグメントを1つ再送し、データセグメント再送を反映するように送信セグメントリストを編集、RTTable [RTC] の値が未設定の場合は再送したデータセグメントのti\_tsva1の値をレコードに設定し（同、ステップY9）、第1の実施の形態のACK処理手段134と同様にエラー制御処理のための各種変数の初期設定を行なう（同、ステップS14）。

## 【 0 0 8 3 】

以降、データセグメント送信処理手段122とACK受信処理手段13はp\_max\_seqで記憶しているデータセグメントが累積確認されるまでエラー制御処理を行なう。

## 【 0 0 8 4 】

エラー制御処理では、ACK処理手段136は、操作する送信セグメントリストが第2の実施の形態の送信セグメントリストに変更されている点を除いては、全て第1の実施の形態のACK処理手段134と同様に処理を行なうためここでは詳細な説明を省略する。

## 【 0 0 8 5 】

エラー制御中に送信要求をされたデータセグメント送信処理手段122は、輻輳ウィンドウpipeの値がsnd\_cwndの値よりも小さい場合にデータセグメントを作成し、送信する（図11中、ステップS40とS50でYes）。その際にはデータセグメント送信処理手段122は、まず、送信セグメントリストから再送すべきデータの有

無を調べ、再送すべきデータがあれば（同、ステップY20でYes）、そのデータセグメントを作成し（同、ステップS52）、生成したデータセグメントに対応するリスト要素を送信セグメントリストから削除し（同、ステップY25）、pipeの値を前記ステップY25で削除したリスト要素が含むデータのサイズ分、すなわち、生成したデータセグメントのサイズ分、減じる（同、ステップX14）。再送すべきデータが無く（同、ステップX12でNo）、送信すべきデータがあれば（同、ステップS53でYes）後続の新規のデータセグメントを作成する（同、ステップS54）。再送すべきデータとはこの場合送信セグメントリストのリスト要素が記憶しているデータで、かつ、リスト要素のTSフィールドの値が現在のラウンドトリップカウンタRTCの値より2小さい値をRTCフィールドに持つRound Tripテーブルのレコードの時刻フィールドの値RTTable [RTC-2] と等しいか、それより小さな値になっているリスト要素が記憶しているデータのことであり、データセグメント送信処理手段122はこのようなデータを送信セグメントリストの先頭方向から順に選択する。すなわち、データセグメント送信処理手段122は、送信セグメントリストの先頭のリスト要素から、 $TS \leq RTTable [RTC-2]$  となるリスト要素を探し、そのようなリスト要素が存在すれば、そのリスト要素のstartからendまでを再送の対象とする。

#### 【 0 0 8 6 】

再送データセグメントの場合も、新規データセグメントの場合もいずれにおいてもpipeの値をそのデータセグメントのサイズ分増やし（同、ステップS55）、そのデータセグメントにタイムスタンプオプションフィールドを付加してそのti\_tsvalフィールドに現在の時刻を設定し（同、ステップY21）、データセグメント送信の内容を示すリスト要素（startとendの値がデータセグメントが含むデータの先頭と末尾のデータのシーケンス番号で、TSの値がデータセグメントti\_tsvalの値）を作成してリスト要素を送信セグメントリストの末尾に追加する（同、ステップY22）。さらに、現時点でのRTCの値をRTCフィールドの値に持つRound Tripテーブルのレコードについて、その時刻フィールドに値が設定されていない場合（同、ステップY23でYes）は、時刻フィールドに前記データセグメントのti\_tsvalの値を設定し（同、ステップY24）、最後に前記データセグメントを送信

する（同、ステップS56）。

【 0 0 8 7 】

以上のように、ラウンドトリップカウンタを導入し、送信装置1でラウンドトリップカウンタと送信したデータセグメントに付加したタイムスタンプオプションのti\_tsvalの値の対応と管理し、送信したデータセグメントのti\_tsvalを記憶し、送受信されるデータセグメントとACKメッセージにタイムスタンプオプションを付加することにより、再送したデータセグメントの損失を検出し、再度、再送することができ、スループット特性を向上させることができる。例えば、既存の技術の課題で説明した図17のシーケンスと同様な状況においては、第1の実施の形態で示した図6で示すのと同様な、シーケンスを実現することができる。

【 0 0 8 8 】

本第2の発明の実施の形態では、データの送信のみを行なうアプリケーションサービス実行手段11を搭載する送信装置1とデータの受信のみを行なうアプリケーションサービス実行手段21を搭載する受信装置2からなる構成を採っているが、第1の実施の形態同様、一つの装置でデータの送信、受信を行なえる装置を構成することができる。

【 0 0 8 9 】

以上説明した本発明によるデータ通信装置（データ通信システム）の主要な特徴をまとめると次のようになる。

【 0 0 9 0 】

本発明の第1の通信システムは、送信装置に、送信するデータパケットのラウンドトリップ数（往復数）を計数するカウンタを保持し、送信するデータパケットにその時点でのカウンタ値をラウンドトリップ通知情報として含ませ、送信した各データパケットに対して、データパケットとデータパケット送信時のカウンタの値の対応を記憶し、記憶したカウンタの値がその時点でのカウンタの値より2つ以上小さい場合に、記憶したカウンタ値に対応するデータパケットが損失したとみなし、データパケットを再送する手段（図1の121）と、受信したデータパケットに対するACKパケットに受信したデータパケットがラウンドトリップ通知情報として含むカウンタ値をラウンドトリップ応答情報として含ませる手段（図

1の221と231) と、受信したACKパケットにラウンドトリップ応答情報として含まれている値がカウンタの値と等しい場合に、カウンタの値を1つ増加させる手段(図1の134)を有する。このような構成を採用し、再送したデータパケットの損失を検出しデータパケットを再送することにより本発明の目的を達成することができる。

#### 【 0 0 9 1 】

本発明の第2の通信システムは、送信するデータパケットのラウンドトリップ数を計数するカウンタを保持し、送信するデータパケットにその時点での時刻を時刻通知情報として含ませ、送信した各データパケットに対して、データパケットとデータパケット送信時の時刻の対応を第1のテーブル(図9のラウンドトリップテーブル)に記憶し、カウンタの取る各カウンタ値に対して、カウンタ値とカウンタがそのカウンタ値を取った後初めてデータパケットを送信するときの送信時の時刻との対応を第2のテーブル(図9の送信セグメントリスト)に記憶し、第1のテーブルに記憶した時刻の値が、第2のテーブルに記憶したその時点でのカウンタの値より2つ小さいカウンタ値に対応する時刻より早い場合に、第1のテーブルに記憶した時刻に対応するデータパケットが損失したとみなし、データパケットを再送する手段(図8の122)と、受信したデータパケットに対するACKパケットに受信したデータパケットが時刻通知情報として含む時刻を時刻応答情報として含ませる手段(図8の222と232)と、受信したACKパケットに時刻応答情報として含まれている時刻が、第2のテーブルに記憶したその時点でのカウンタの値と対応させて記憶させた時刻と等しいかまたは大きい場合に、カウンタの値を1つ増加させる手段(図8の136)を有する。このような構成を採用し、再送したデータパケットの損失を検出しデータパケットを再送することにより本発明の目的を達成することができる。

本発明の第3の通信システムは、エラー制御中にデータ受信装置から受信したACKパケットで新規に送達確認がなされたデータの合計サイズ分の閉じられたウィンドウを送信ウィンドウから解放し、その分を送信可能とする手段(図1の133と134。図8の135と136)。このような構成を採用することにより、本発明の目的

を達成することができる。

#### 【0092】

なお、本発明の実施の形態は、上述したものに限定されることなく、例えば、各装置および各装置内の各構成要素を統合、分割したり、分散して配置する等、適宜変更可能である。また、本発明のデータ通信装置あるいはデータ通信システムは、送信部分と受信部分を分割した上で製品として頒布することが可能である。また、本発明の通信装置あるいは送信装置もしくは受信装置は、コンピュータおよびその周辺装置と、コンピュータによって実行されるプログラムとから構成することができ、そのコンピュータによって実行れるプログラムも、全体あるいは部分として、コンピュータ読み取り可能な記録媒体または通信回線を介して頒布することおよび利用することが可能である。

#### 【0093】

##### 【発明の効果】

第1の効果は、再送されたデータセグメントが損失した場合において、既存の技術よりも優れたスループット特性を得られることである。その理由は、送信するデータパケットのラウンドトリップ数を計数するカウンタ (Round Trip Counter) を導入し、再送時に損失したデータセグメントに対しては、Round Trip Counterの値が2以上増えても送達が確認されないことによりデータセグメントの損失を検出し、データセグメントの再送が行なえるからである。

#### 【0094】

第2の効果は、エラー制御処理中に送信されたデータセグメントの到着順序が入れ替わった場合、または、送信ウィンドウサイズが十分大きくない場合において、既存の技術に比べスループット特性が向上することである。その理由は、データセグメントを再送するか、否かの判断に、送信ウィンドウサイズとは関係の無いラウンドトリップ数 (Round Trip数) を用いているからである。Round Trip数を用いた再送判断はデータセグメントの到着順序の入れ替わりに対して完全に対応できるものではないが、同一Round Tripのデータセグメントの到着順序の入れ替わりに対しては無駄な再送が発生せず、この場合は既存の技術に比べてスループット特性が向上する。

【 0 0 9 5 】

第3の効果は、エラー制御処理中にデータ受信装置（例えば受信側TCP）からのACKメッセージが損失した場合において、既存の技術に比べスループット特性が向上することである。その理由は、エラー制御処理中のネットワークに滞留しているデータサイズの推定の際に、例えばSACKオプションと累積確認で送達が確認されたデータのサイズ、すなわちACKパケットで新規に送達確認がなされたデータの合計サイズ分を考慮し、送信フロー制御を行なうための送信ウィンドウのうちの閉じられたウィンドウを送信ウィンドウから解放してその分を送信可能としているからである。

【図面の簡単な説明】

【図1】 本発明の第1の実施の形態の構成を示すブロック図

【図2】 本発明の第1の実施の形態のデータ構造を示す図

【図3】 RTCオプションフィールドの構成の例を示す図

【図4】 本発明の第1の実施の形態の送信動作を示す処理フロー図

【図5】 本発明の第1の実施の形態の受信動作を示す処理フロー図

【図6】 本発明の第1の実施の形態の動作の例を示すシーケンス図

【図7】 本発明の第1の実施の形態の動作の例を示すシーケンス図

【図8】 本発明の第2の実施の形態の構成を示すブロック図

【図9】 本発明の第2の実施の形態のデータ構造を示す図

【図10】 タイムスタンプオプションフィールドの構成を示す図

【図11】 本発明の第2の実施の形態の送信動作を示す処理フロー図

【図12】 本発明の第2の実施の形態の受信動作を示す処理フロー図

【図13】 既存の技術の構成を示すブロック図

【図14】 既存の技術のデータ構造を示す図

【図15】 既存の技術の送信動作を示す処理フロー図

【図16】 既存の技術の受信動作を示す処理フロー図

【図17】 既存の技術の動作の例を示すシーケンス図

【図18】 既存の技術の動作の例を示すシーケンス図

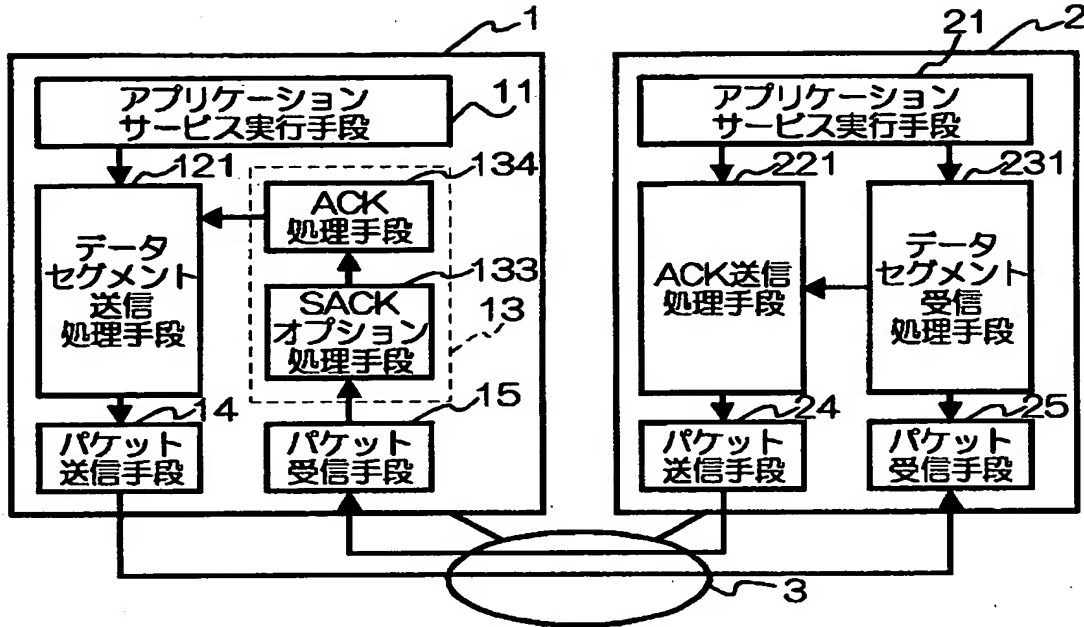
【符号の説明】

- 1 送信装置
- 2 受信装置
- 3 ネットワーク
- 11 アプリケーションサービス実行手段
- 12 データセグメント送信処理手段
- 13 ACK受信処理手段
- 14 パケット送信手段
- 15 パケット受信手段
- 21 アプリケーションサービス実行手段
- 22 ACK送信処理手段
- 23 データセグメント受信処理手段
- 24 パケット送信手段
- 25 パケット受信手段
- 121 データセグメント送信処理手段
- 122 データセグメント送信処理手段
- 131 SACKオプション処理手段
- 132 ACK処理手段
- 133 SACKオプション処理手段
- 134 ACK処理手段
- 135 SACKオプション処理手段
- 136 ACK処理手段
- 221 ACK送信処理手段
- 222 ACK送信処理手段
- 231 データセグメント受信処理手段
- 232 データセグメント受信処理手段

【書類名】

図面

【図1】

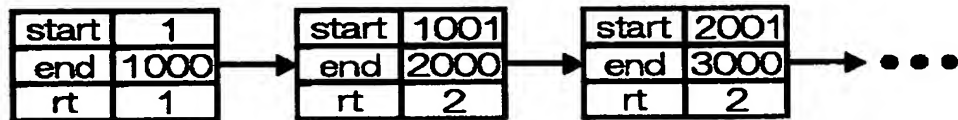


【図2】

ラウンドトリップカウンタ

RTC 2

送信セグメントリスト



【図 3】

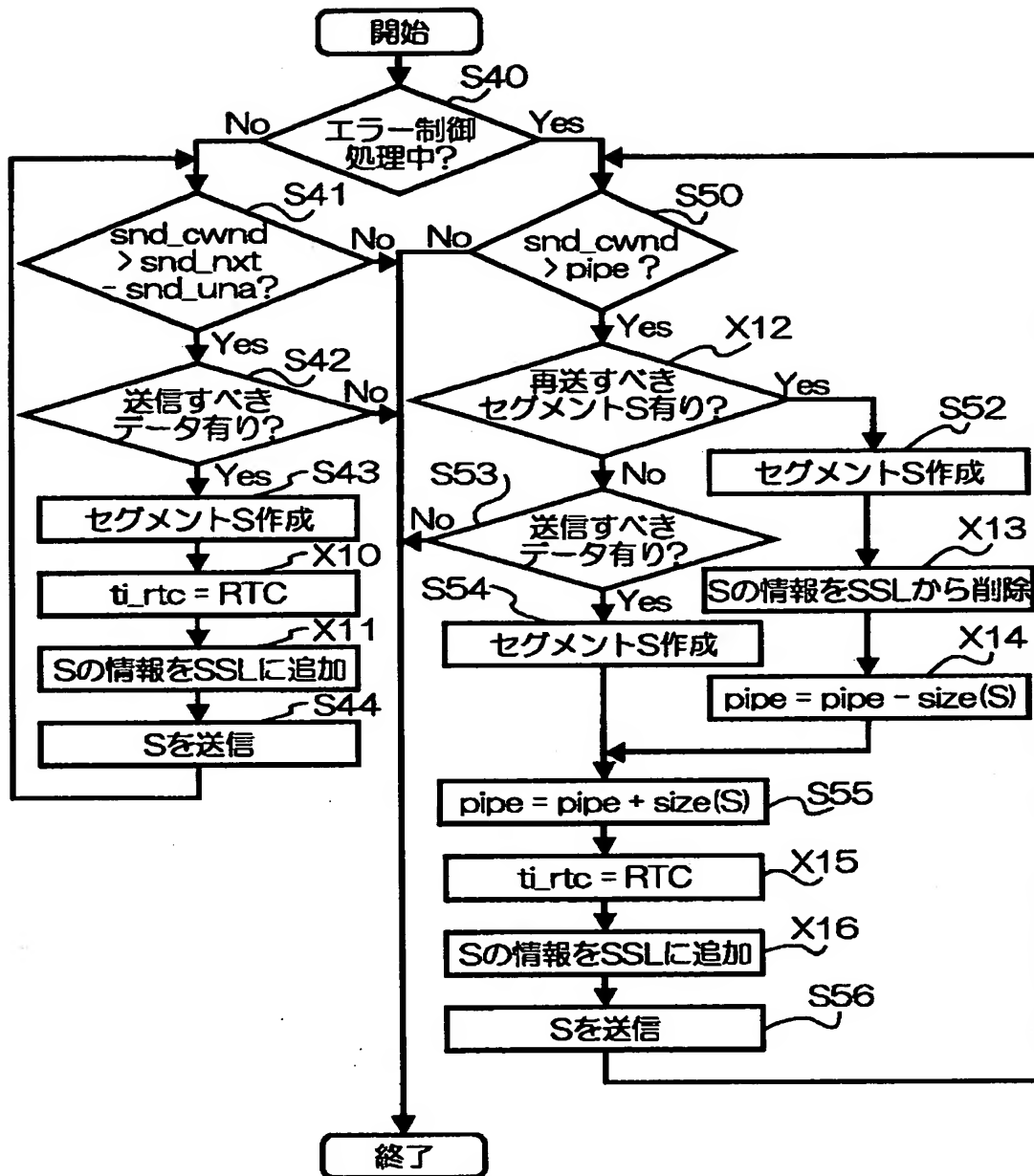
ラウンドトリップカウンタ・オプション

0	7 8	(bit) 15
Kind="Round Trip Counter"		Length=3
Round Trip Counter 値		

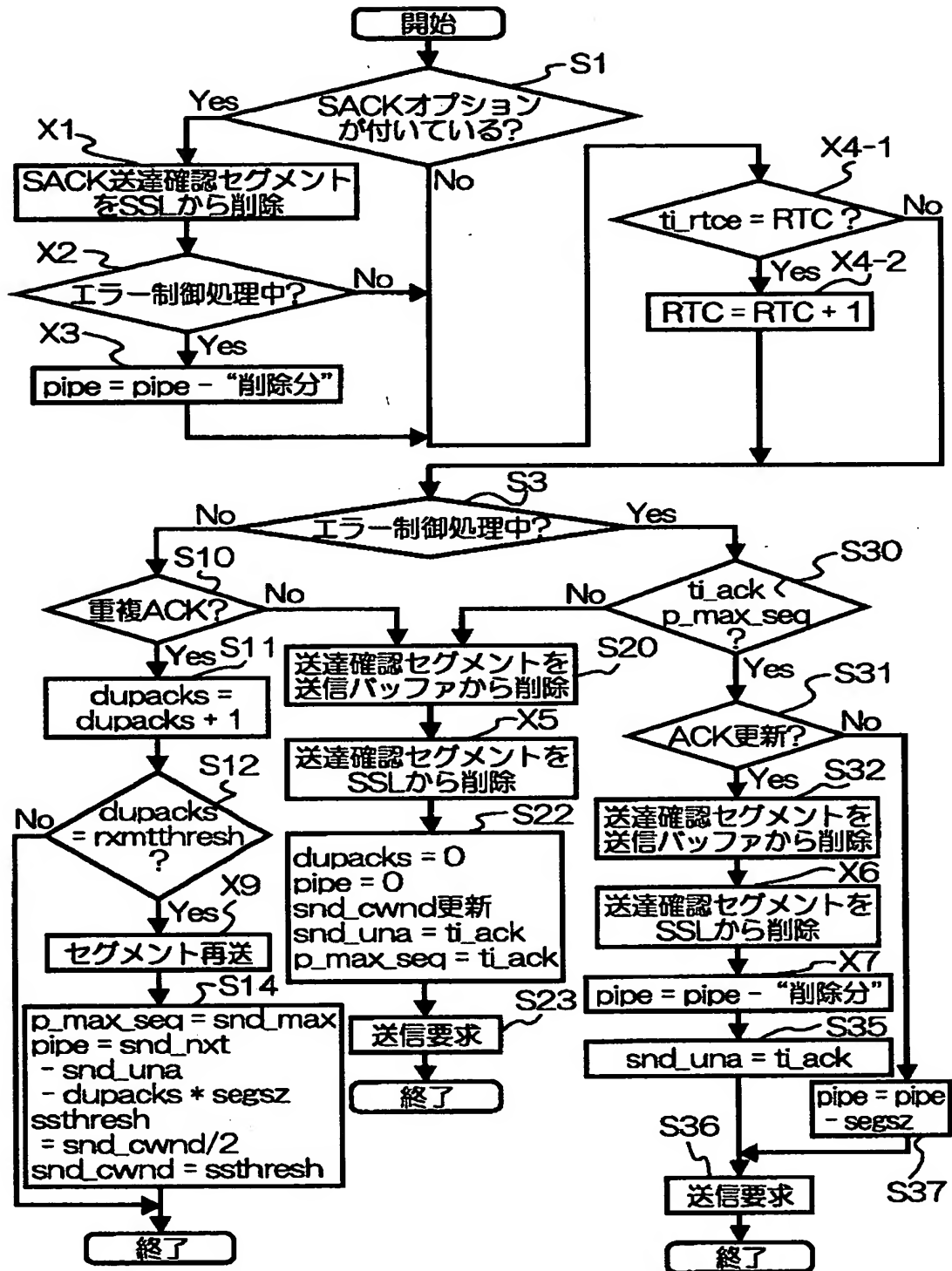
ラウンドトリップカウンタエコー・オプション

0	7 8	(bit) 15
Kind="Round Trip Counter Echo"		Length=3
Round Trip Counter Echo 値		

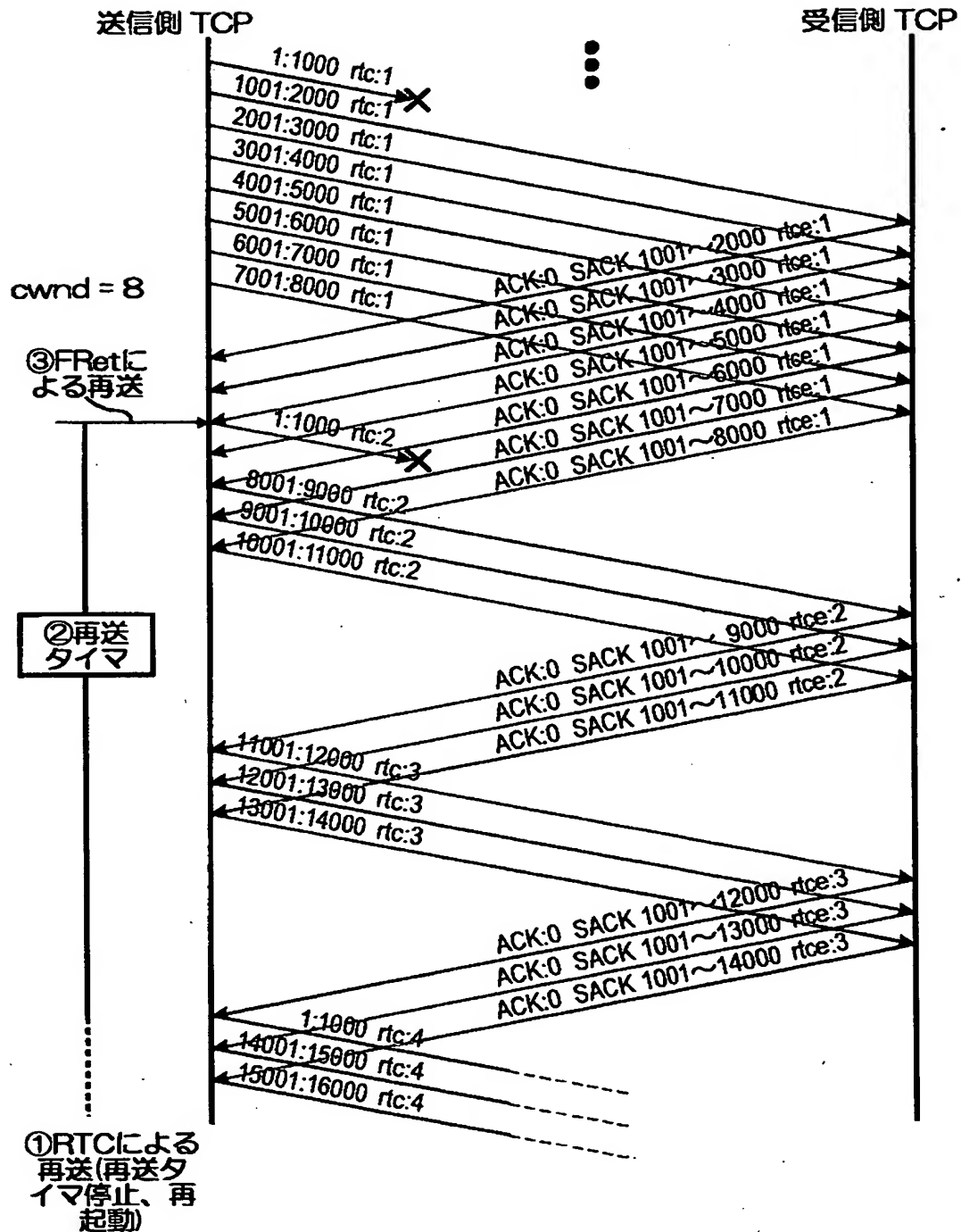
【図 4】



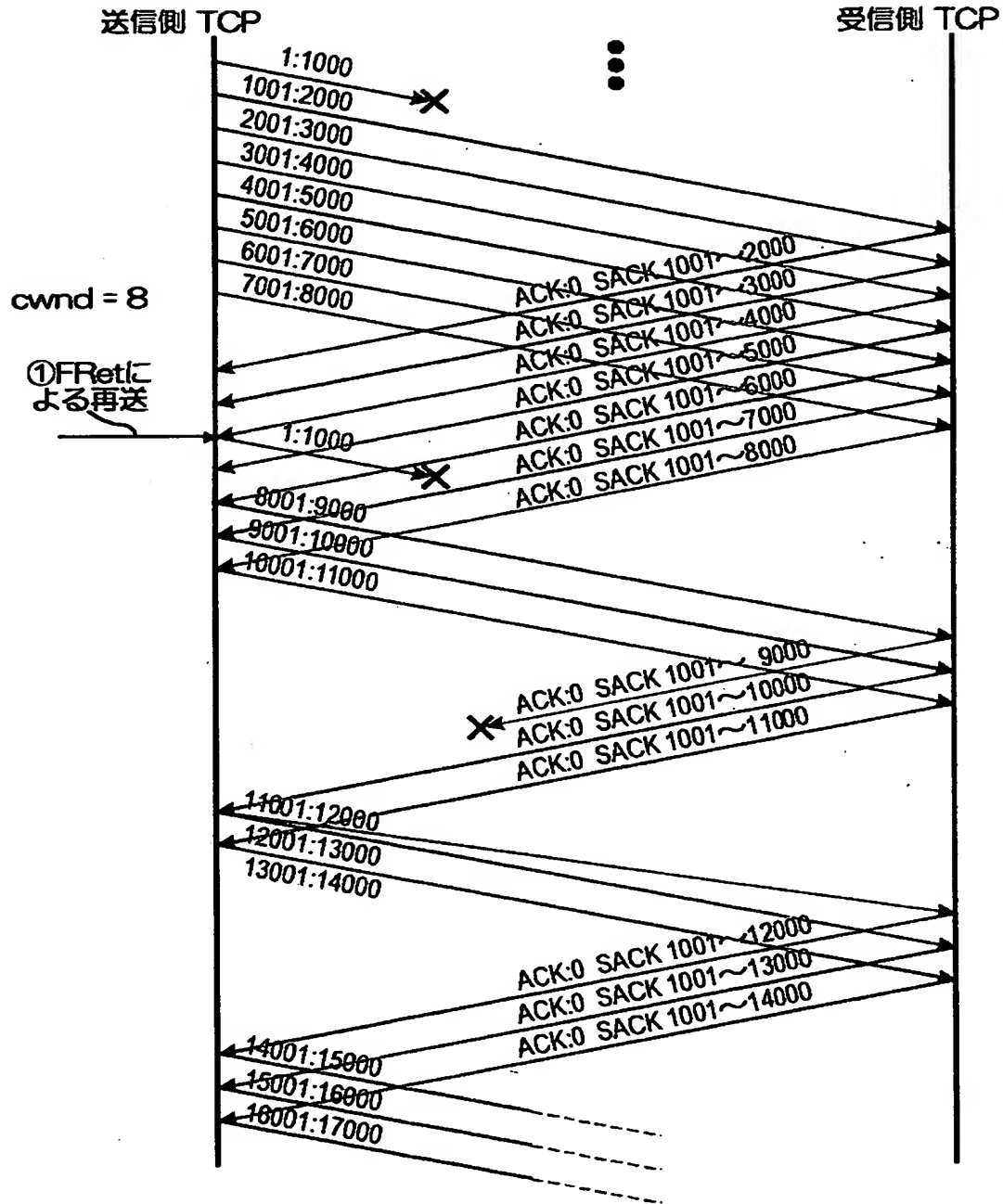
【図 5】



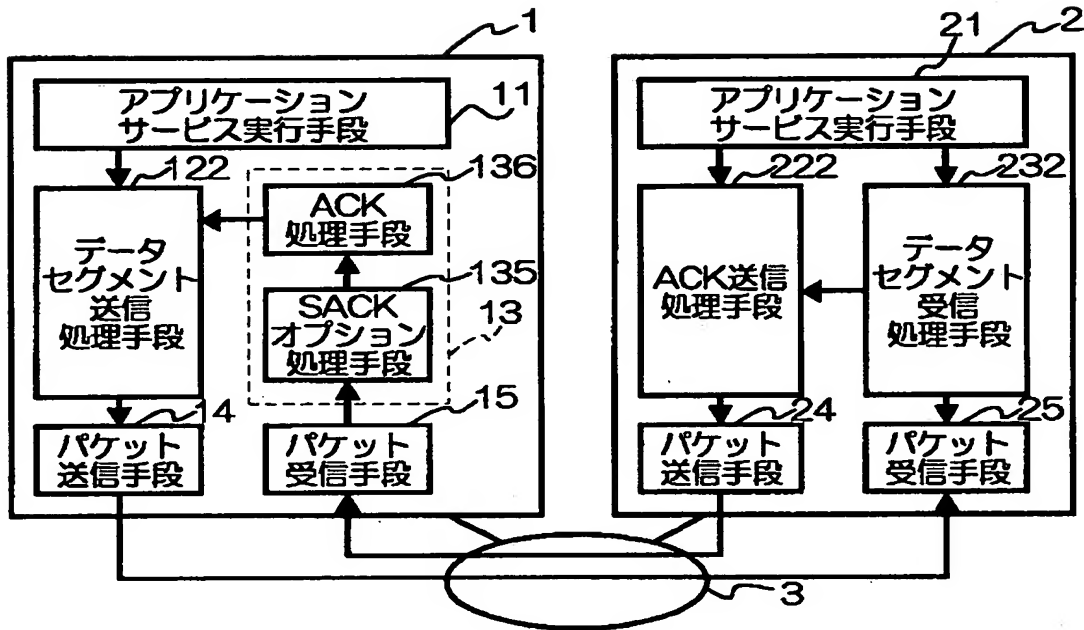
【図 6】



【図 7】



【図 8】



【図 9】

ラウンドトリップカウンタ

RTC	2
-----	---

ラウンドトリップテーブル

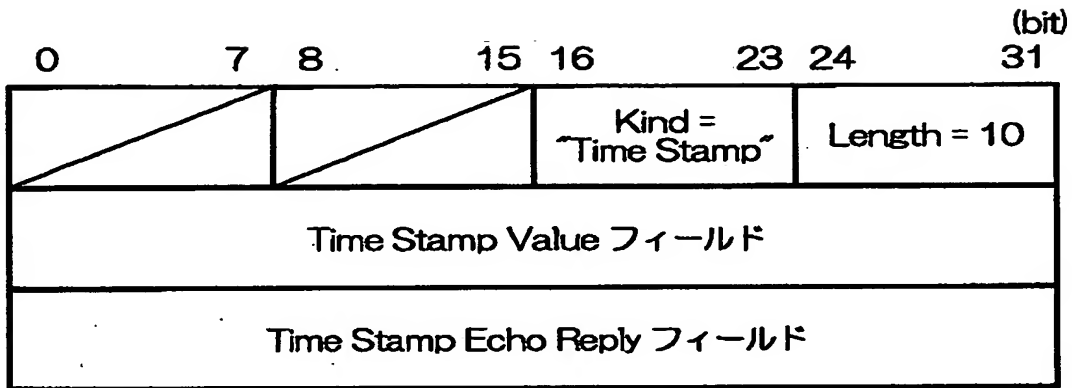
RTC	時刻
1	11111
2	11222
⋮	⋮

送信セグメントリスト

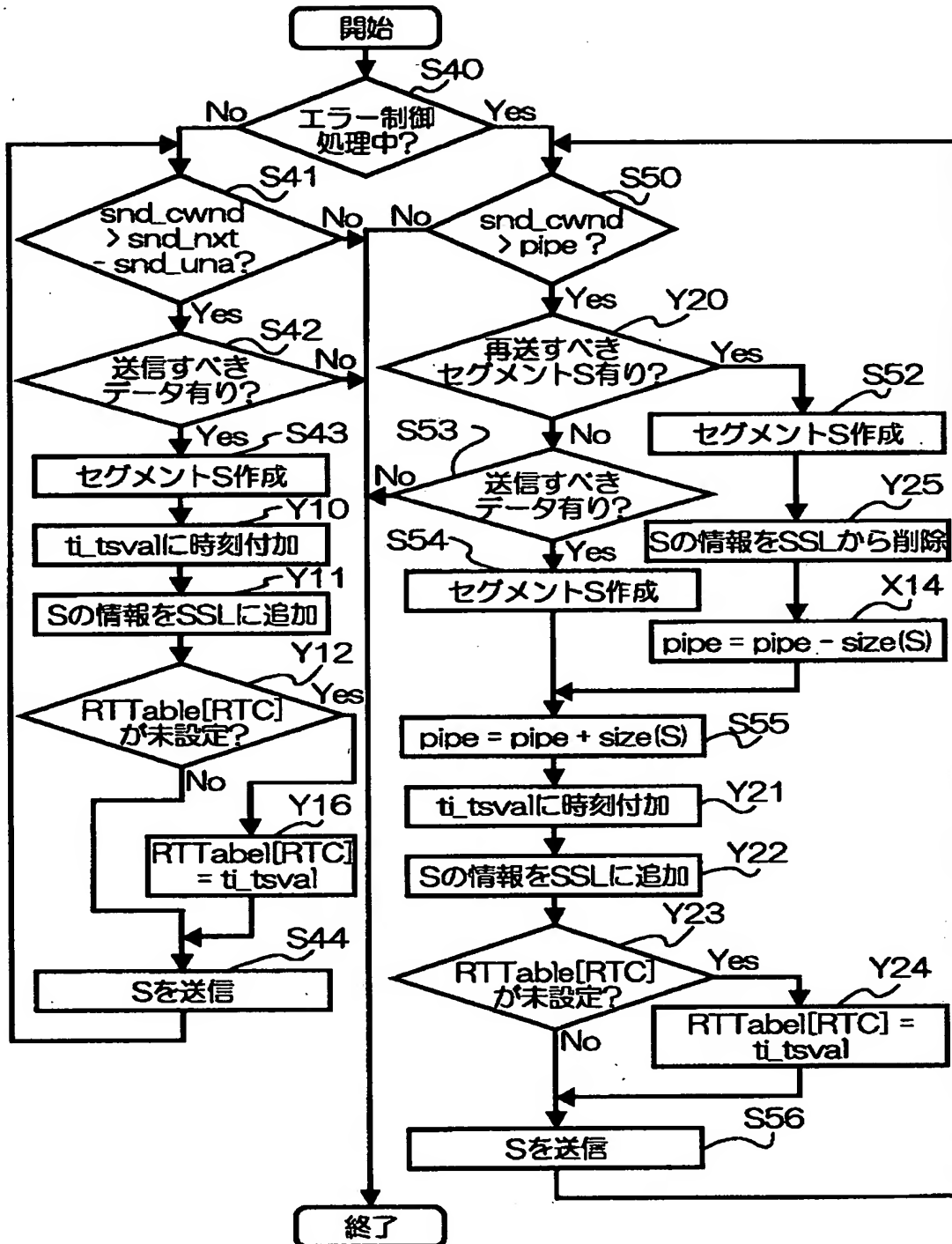
start	1	→	start	1001	→	start	2001	→	...
end	1000		end	2000		end	3000		
TS	11111		TS	11222		TS	11333		

【図 1 0】

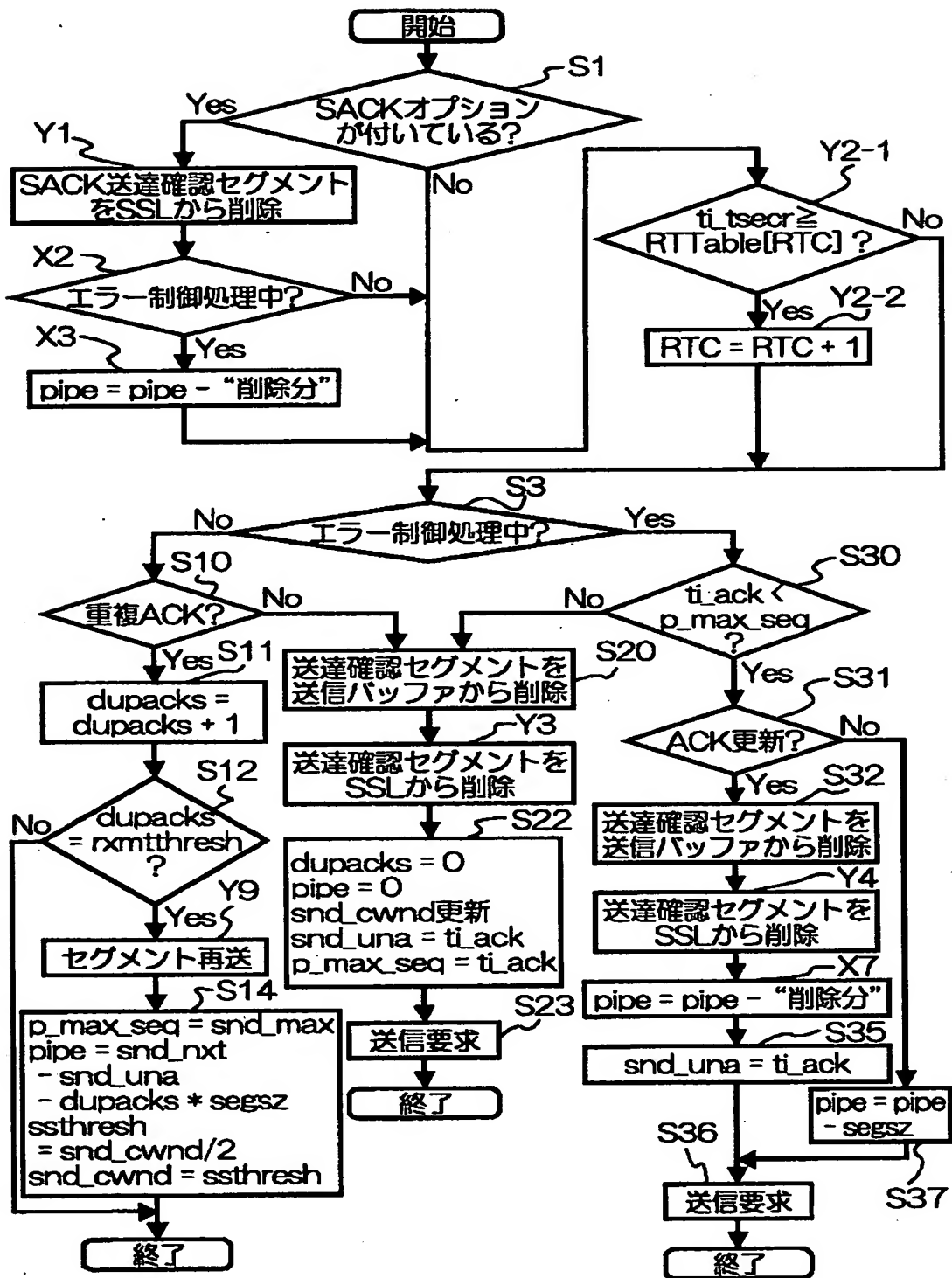
タイムスタンプ・オプション



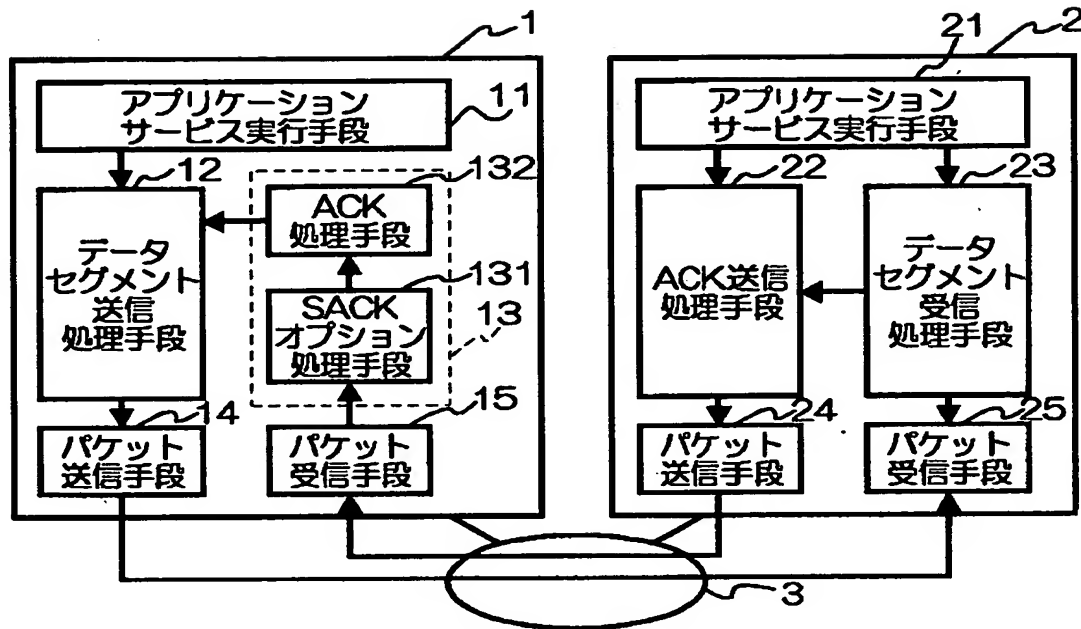
【図11】



【図12】

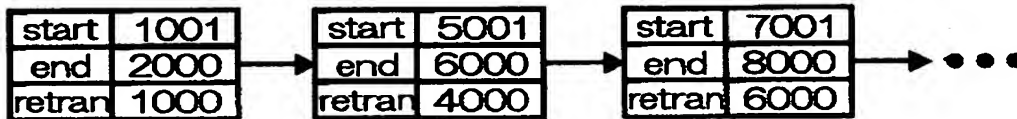


【図 13】

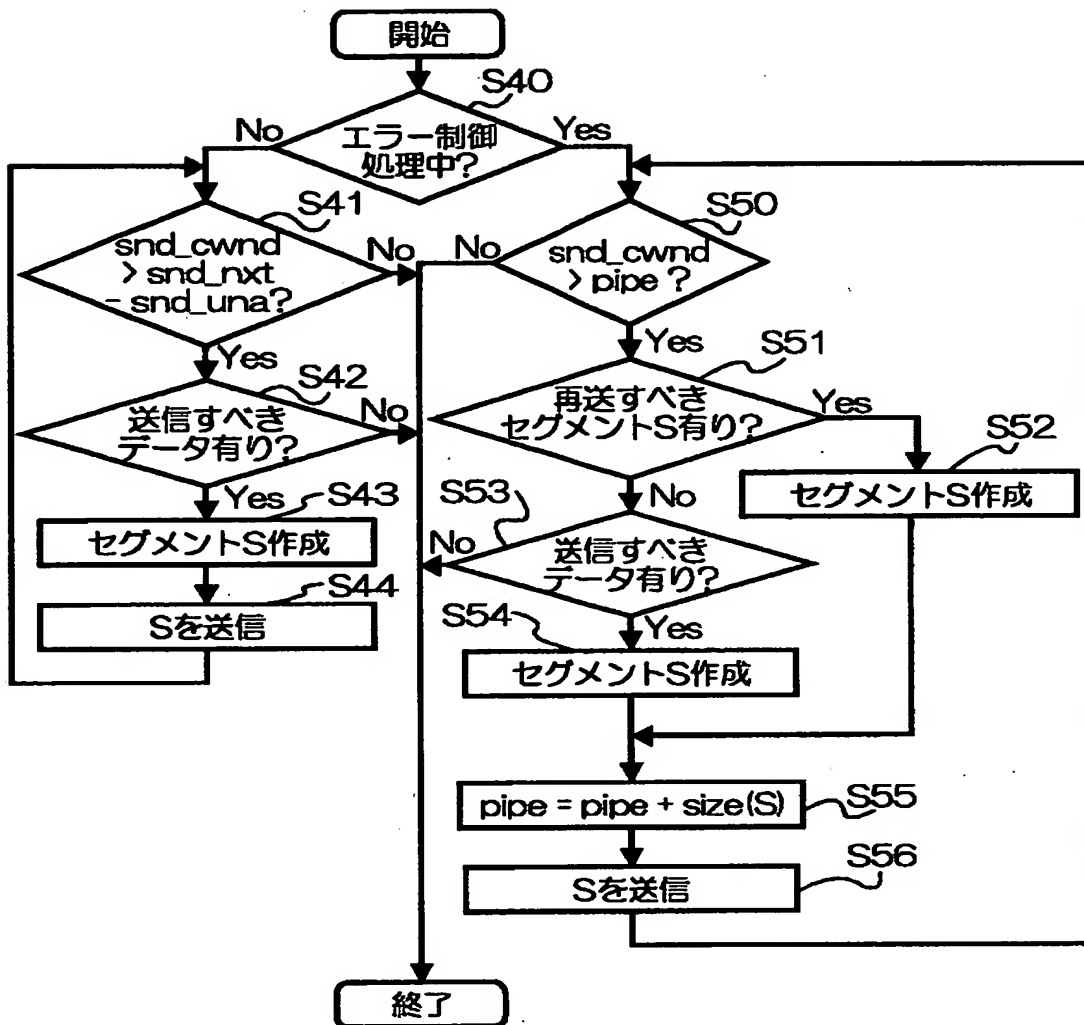


【図 14】

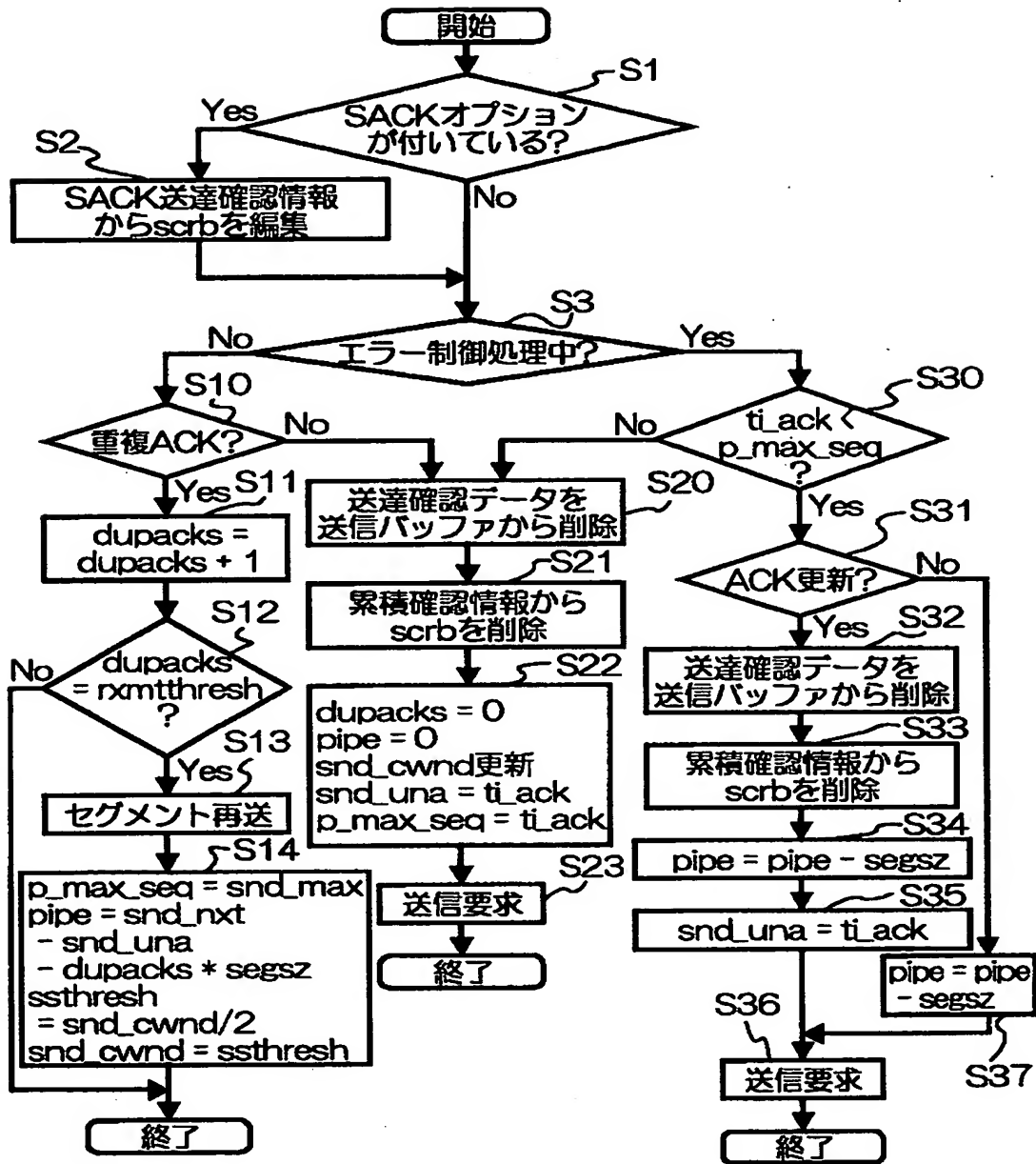
スコアボード



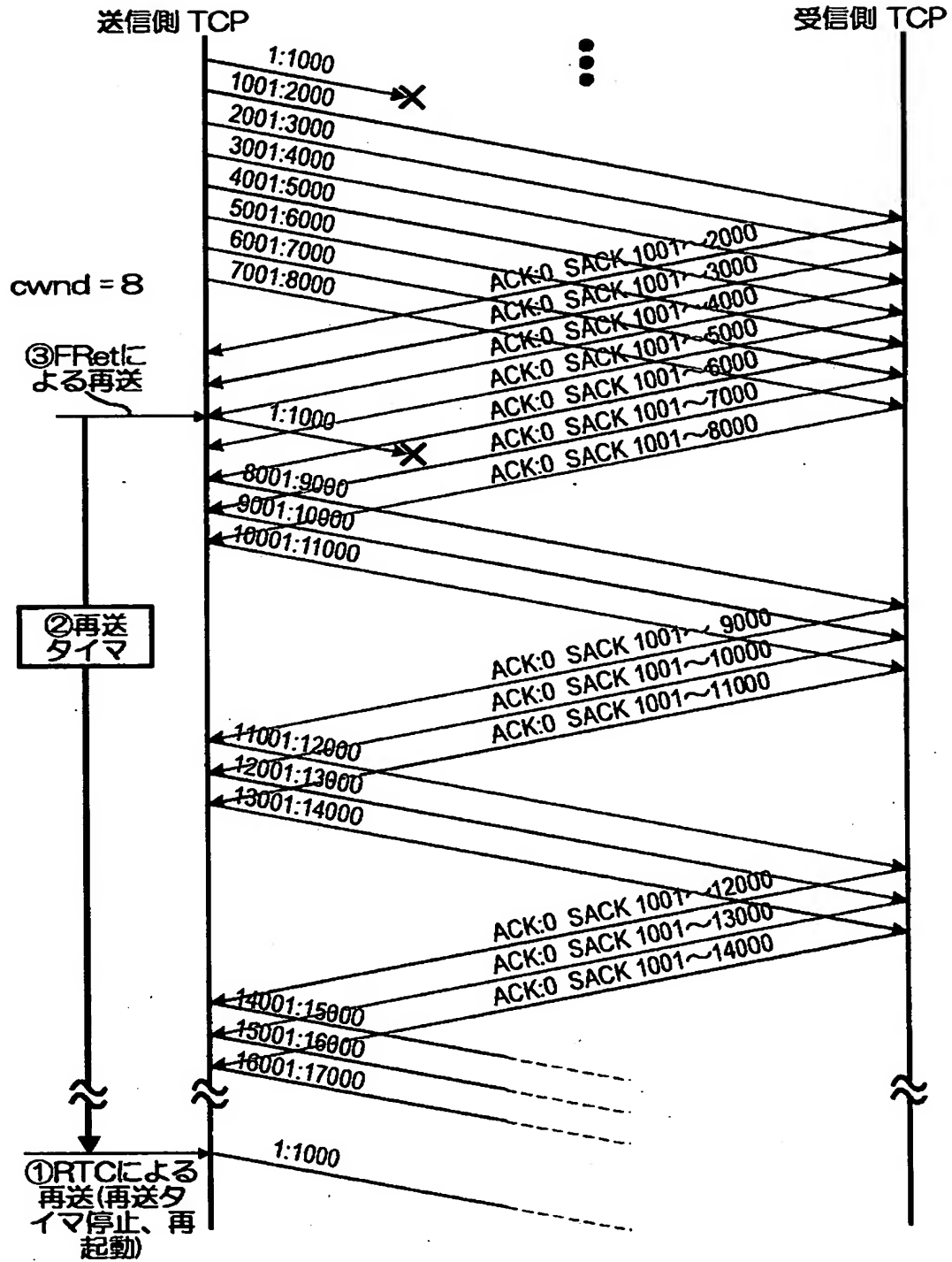
【図15】



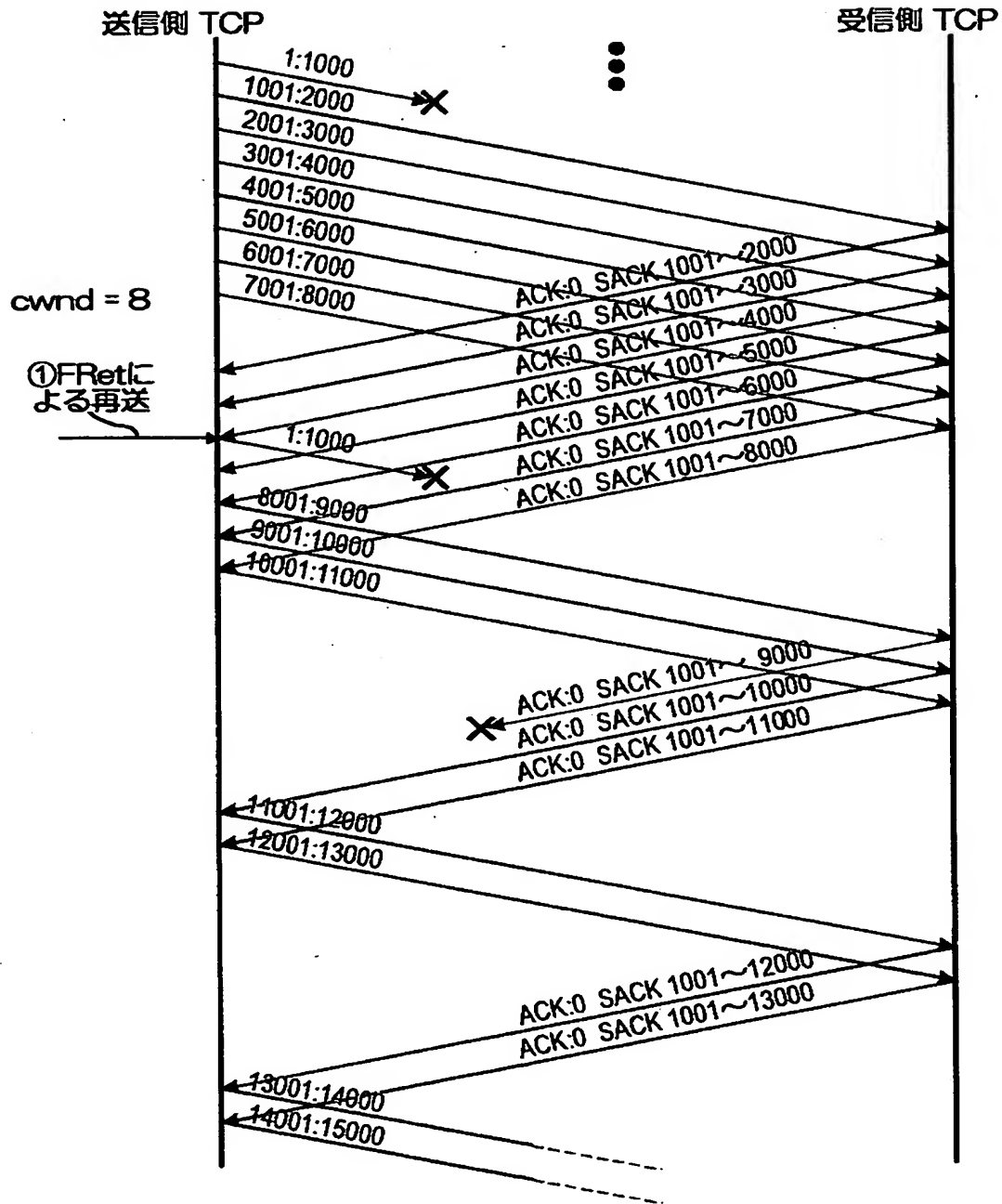
【図16】



【図 17】



【図18】



【書類名】 要約書

【要約】

【課題】 再送したデータセグメントが損失しても送信装置と受信装置間のスループット劣化を回避する。

【解決手段】 送信装置1と受信装置2とネットワーク3からなる通信システムで送信装置1に送信データセグメントのラウンドトリップ数を計数するカウンタを導入する。データセグメント送信処理手段121は送信するデータセグメントにカウンタ値を付加して送信し、ACK送信手段221は受信したデータセグメントに含まれたカウンタ値をACKメッセージに付加して返信し、ACK処理手段134は受信したACKメッセージに付加されたカウンタ値が現在のカウンタ値と同じ値の場合にカウンタ値を1つ増やすことでラウンドトリップ数を計数する。データセグメント送信処理手段121は、データセグメント送信時にデータセグメントと付加したカウンタ値の対応を記憶し、記憶したカウンタ値がその時点のカウンタ値よりも2以上小さい場合、データセグメントを再送する。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願2000-341680
受付番号	50001447700
書類名	特許願
担当官	佐藤 一博 1909
作成日	平成12年11月16日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000004237
【住所又は居所】	東京都港区芝五丁目7番1号
【氏名又は名称】	日本電気株式会社

【代理人】

【識別番号】	100108578
【住所又は居所】	東京都新宿区高田馬場3丁目23番3号 ORビ ル 志賀国際特許事務所
【氏名又は名称】	高橋 詔男

【代理人】

【識別番号】	100064908
【住所又は居所】	東京都新宿区高田馬場3丁目23番3号 ORビ ル 志賀国際特許事務所
【氏名又は名称】	志賀 正武

【選任した代理人】

【識別番号】	100101465
【住所又は居所】	東京都新宿区高田馬場3丁目23番3号 ORビ ル 志賀国際特許事務所
【氏名又は名称】	青山 正和

【選任した代理人】

【識別番号】	100108453
【住所又は居所】	東京都新宿区高田馬場3丁目23番3号 ORビ ル 志賀国際特許事務所
【氏名又は名称】	村山 靖彦

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日	1990年 8月29日
[変更理由]	新規登録
住 所	東京都港区芝五丁目7番1号
氏 名	日本電気株式会社